VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

PREDICTION AND ANALYSIS OF NUCLEOSOME POSITIONS IN DNA

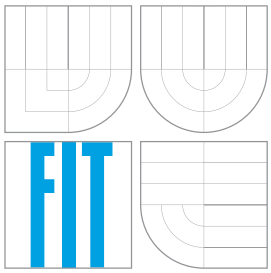
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MAREK VIŠŇOVSKÝ

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

PREDIKCE A ANALÝZA POZIC NUKLEOZOMŮ V DNA

PREDICTION AND ANALYSIS OF NUCLEOSOME POSITIONS IN DNA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MAREK VIŠŇOVSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ MARTÍNEK, Ph.D.

BRNO 2013

Abstrakt

Eukaryotní DNA se váže kolem nukleozomů, čím ovplyvnuje vyšši strukturu DNA a přístup k vazebním mistům pro všeobecní transkripční faktory a oblasti genů. Je proto důležité vědet, kde se nukleozomy vážou na DNA, a jak silná tato vazba je, abychom mohli porozumět mechanizmům regulace genů. V rámci projektu byla implementována nová metoda pro predikci nukleozomů založená na rozšíření Skrytých Markovových modelů, kde jako trénovací a testovací sada posloužila publikována data z Brogaard et al. (Brogaard K, Wang J-P, Widom, J. Nature 486(7404), 496-501 (2012). doi:10.1038/nature11142). Správne predikováno bylo zhruba 50% nukleozomů, co je porovnatenlný výsledek s existujícimi metodami. Okrem toho byla provedena řada experimentů popisující vlastnosti sekvencí nukleozomů a ich organizace.

Abstract

Genomic DNA in eukaryotes wraps around nucleosomes, which thereby affects higher order DNA structure and access to genomic features like transcription factor binding sites (TFBSs) and gene regions. It is therefore important to have a good understanding of where nucleosomes bind to DNA, and how stable this binding is, in order to understand gene regulation. We developed, implemented and tested a novel approach for genome-wide predictions of nucleosome positions in yeast based on Hidden Markov models extended by duration modeling, using data from Brogaard et al. (Brogaard K, Wang J-P, Widom, J. Nature 486(7404), 496-501 (2012). doi:10.1038/nature11142) for training and testing. Achieved sensitivity closing to 50% does not improve performance compared to other existing methods. In addition, several experiments were conducted on the available dataset to identify features of sequences occupied by nucleosomes and theirs global organization that are important for the prediction performance.

Klíčová slova

strojové učení, predikce pozic nukleozomů, Skryté Markovovy modely

Keywords

machine learning, nucleosome positioning prediction, Hidden Markov Models

Citace

Marek Višňovský: Prediction and analysis of nucleosome positions in DNA, diplomová práce, Brno, FIT VUT v Brně, 2013

Prediction and analysis of nucleosome positions in DNA

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Tomáše Martínka Ph.D.

.....
Marek Višňovský
July 24, 2013

Poděkování

Táto práca je výsledkom výmenného študijného pobytu v rámci programu ERASMUS a bola vypracovaná z veľkej časti na NTNU v Trondheime. Moja veškerá vďaka patrí ľuďom, ktorí toto všetko umožnili, menovite profesorovi Finn Drabløs, ktorý navrhol zadanie a ponúkol odborný pohľad na danú problematiku, profesorovi Pål Sætrom, ktorý sa postaral o všetku administratívu na strane NTNU a v neposlednom rade môjmu školiteľovi na FIT VUT Ing. Tomášovi Martínkovi Ph.D. za pomoc a obstaranie administratívnych záležitostí na strane VUT.

Contents

List of Figures

Chapter 1

Introduction

Genomic DNA in eukaryotes wraps around nucleosomes, which thereby affects higher order DNA structure and access to genomic features like transcription factor binding sites (TFBSs) and gene regions. It is therefore important to have a good understanding of where nucleosomes bind to DNA, and how stable this binding is, in order to understand gene regulation.

Existing methods for nucleosome positioning prediction are mostly based on sequence dependent features of DNA occupied by nucleosomes or not (linker DNA). In general, there are two key properties of sequences favored by nucleosomes to capture — the periodic occurrences of particular dinucleotides and k-mer composition preferences, which slightly differ from those observed in linker sequences. The performance of these methods is limited by the fact that the nucleosome positioning *in vivo* is influenced not only by intrinsic organization of underlying sequence, but also by the activity of environmental factors within the cell such as actions of chromatin remodelers or competition with site specific DNA-binding proteins. Another downside of existing methods is a shortage of accurate nucleosome positioning data, which is to be changed by recent works that made finally complete genome maps of nucleosome positions available with high accuracy.

The main purpose of this thesis is to develop a machine learning technique for prediction of nucleosome positions in DNA, focusing on the yeast as model organism for which we have the most accurate map of nucleosome positions available to this date [4].

The chapter 2 serves as an introduction to the biology behind the DNA packaging and the chromatin structure, focusing on the role of nucleosomes within. In the chapter 3 we turn our attention to existing methods, describing their principles and computational approaches, they are based on. The next chapter, chapter 4, presents analysis and experiments conducted on our dataset. Moreover, it includes a proposal of novel prediction method based on presented observations and situated within Hidden Markov model framework. The chapter 5 focuses on the testing of proposed approach, evaluation of its performance and discussion concerning achieved results as well as brief comparison with one of existing methods. The last chapter, chapter 6, summarize the work related to this thesis and offers some suggestions for the future development.

Chapter 2

Biology of chromatin

Typical length of genomic DNA is hundreds times larger than the diameter of cell's nucleus. In order to fit such large molecule into relatively small nucleus, DNA has to be packed into more condensed form. This task depends on specialized proteins that bind to the DNA and compress it. Interestingly, DNA is packed in such way, which allows it to interact with enzymes and proteins necessary for its transcription, replication and repair.

In eukaryotes, DNA stored in nucleus is divided into the set of different chromosomes. Each chromosome consists of one long linear DNA molecule and proteins bound to this molecule, which winds DNA into more compact form in complex called chromatin. The packaging of DNA in chromatin plays three important roles within the cell: protection against chemical and physical damage that could be lethal to the cell in many cases, compaction and DNA metabolism, when chromosomes serve as platforms for key cell processes.

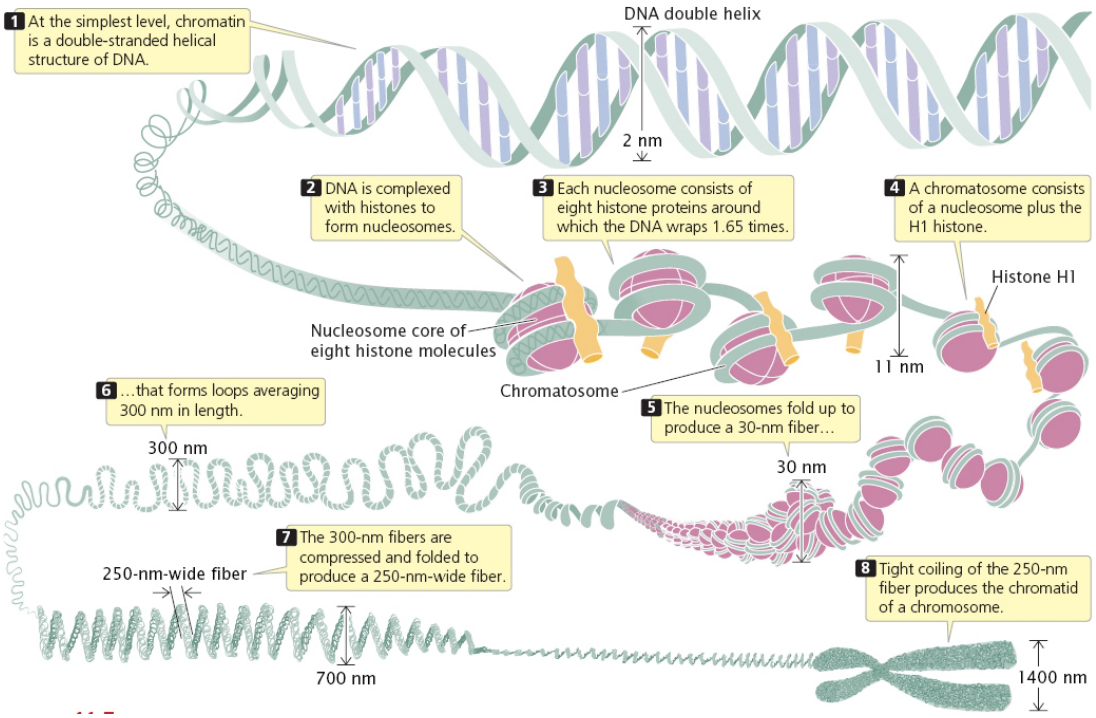


Figure 2.1: Levels of chromatin organization. Source: [1]

Several degrees of DNA packaging are found within an eukaryotic nucleus. At the lowest level, we have a basic unit of chromatin, nucleosome. Nucleosome core comprises of complex of eight histones — two molecules of each of histones H2A, H2B, H3 and H4 — and DNA chain approximately 147 base pairs long. Histones of nucleosome core are evolutionary the most conserved proteins in eukaryotes, which underlines their crucial role in chromatin. These nucleosomes occur in DNA roughly every 200 base pairs (147bp wrapped around histone core and about 50bp of so called linker DNA that connects adjacent nucleosome cores) and are the reason of „beads on a string“ appearance of chromatin fibers. Another protein from histone family, histone H1, mediates compaction of chromatin fiber into 30nm fiber.

Degree of DNA compaction also varies according to the stage of the cell cycle, for example, prior to cell division, DNA forms highly condensed structures suitable for physical separation at mitosis.

2.1 Nucleosome as fundamental unit of chromatin

As mentioned earlier, nucleosome core consists of eight 'core' histones. Histones are small proteins rich in positively charged amino acids. They exhibit a similar polypeptide chain fold — histone fold, based on a long central alpha helix, flanked on both sides by shorter helices and loops that interact with DNA. Outside of the histone fold, the N-terminus of the histone forms an unstructured 'tail', very rich in lysine residues that serves as targets for various secondary modifications, such as methylation or acetylation, which play a key role in chromatin regulation.

Approximately 147 base pairs of DNA wraps in 1.7 turn around nucleosome core. Contact between DNA and protein core is between the negatively charged DNA backbone and the positively charged histone proteins. Nucleosomes are separated by short stretches of DNA. This linker DNA is a crucial determinant in the structure and compaction of the chromatin fiber.

Chromatin not only serves as a way to condense DNA within the cellular nucleus, but also participate on the usage of DNA. Both, interactions between histone fold and DNA in the core particle of nucleosomes and histone tail interactions in the chromatin fiber contribute to gene repression, and each is counteracted by specific mechanism.

First of all, nucleosomes occlude their wrapped DNA, which results in inaccessibility of functional DNA binding sites. For example, packaging promoters in nucleosomes prevents the initiation of transcription by RNA polymerases and transcription factors that leads to gene repression. On the other hand, occurrence of nucleosomes does not stop elongation step of transcription as RNA polymerase is able to move along nucleosomes and temporarily disrupt their structure. Repression due to interactions in the core particle is opposed by process called chromatin remodeling. Chromatin remodelers are large, multiprotein complexes that use the energy of ATP hydrolysis to move nucleosomes along the DNA strand and thus are able to provide access to the underlying DNA to enable transcription.

Secondly, differing levels of acetylation and methylation on histone tails are linked to altered rates of DNA. Repression due to condensation in the chromatin fiber may be relieved by acetylation of histone tails and reestablished by deacetylation. Different types of modifications on histones have been called histone code.

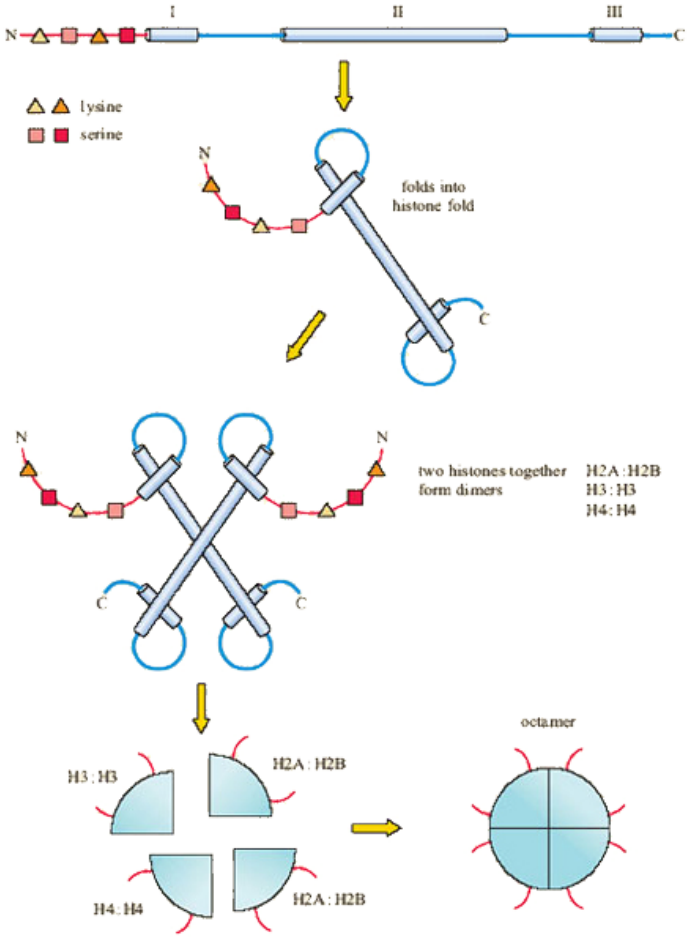


Figure 2.2: Structure of the histone fold and an assembly of histone octamer. Domains I, II and III represent α helices. Source: [22]

2.1.1 Nucleosome sequence properties

Although the interaction between the DNA and histones is not sequence-specific, the sequence of DNA affects its ability to form nucleosomes.

The most notable DNA sequence motif is periodic occurrence of AA/TT/AT/TA dinucleotides every 10.5 base pairs, which are favored when the DNA backbone (minor groove) faces inwards towards the histone core, offset by 5 base pairs from a similarly repeating CC/GC/CG/GG dinucleotides, when the DNA backbone faces outwards.

On other hand, poly(dA:dT) tracts are known for excluding nucleosomes.

2.1.2 Global nucleosome organization

One of the main descriptors of global nucleosome organization is the length of linker DNA between adjacent nucleosomes. This length varies not only between different species, but also different tissues. The shortest linkers are in yeast and fly, with lengths of 20 - 30bp in average, human chromatin has longer linkers with lengths around 40bp and the longest known linker were found in echinoid sperm up to 90bp long.

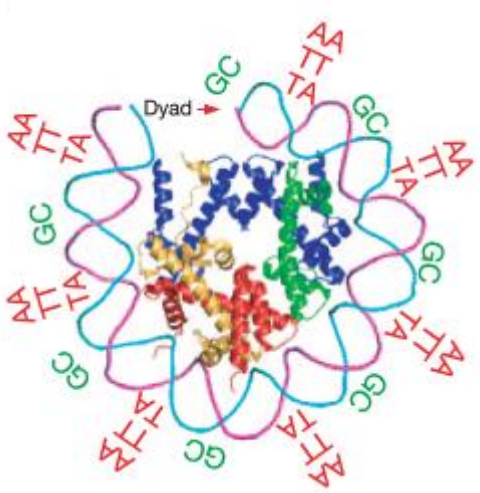


Figure 2.3: Periodic motif of specific dinucleotides in DNA wrapped around nucleosome core. Source: [11].

Nucleosome positioning may also be affected by their chromosomal location, while centromers and gene coding regions are considered densely occupied, nucleosome occupancy of telomeres and intergenic regions is more sparse.

Specific pattern in nucleosome organization is mainly related to open promoters that are accessible by transcriptional machinery. These promoters consist of nucleosome depleted region upstream of transcriptional start site, flanked by two very well positioned nucleosomes, +1 located downstream and -1 upstream. TSS tends to be positioned just on the edge of the +1 nucleosome [2].

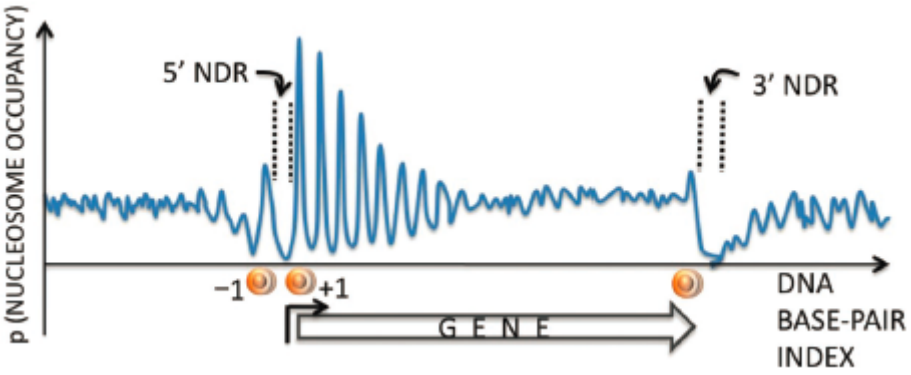


Figure 2.4: Common nucleosome organization along gene coding regions — nucleosome depleted region around promoter flanked by two strongly positioned nucleosomes, which cause equal phasing decaying with distance within the gene-coding region and another nucleosome depleted region downstream the termination site. Source: [2].

Furthermore, nucleosomes located downstream exhibit strong phasing that decays with distance from the TSS. This phasing is probably dictated by statistical positioning principles [14, 5], when the presence of physical barrier influences the position of the neighbouring

nucleosome, which then acts as a barrier for the next nucleosome and so on. These positioning restrictions decrease with the distance from the original barrier. In addition, this effect is predicted to get stronger with increasing nucleosome density, which might altogether explain nucleosome phasing downstream the +1 nucleosome.

Chapter 3

Methods for nucleosome prediction

Majority of methods for nucleosome prediction is based on two key features of DNA sequence occupied by nucleosomes — periodicity of particular dinucleotides and the difference in k-mer usage preferences between nucleosomal and linker DNA. Another common feature of these methods is an usage of Hidden Markov models for genome-wide predictions.

This chapter provides a brief overview of tools available for nucleosome positioning prediction and describes principles that these method are based on.

3.1 Hidden Markov Models

Hidden Markov Models are known mainly from speech signal processing field [16], where are successfully applied for instance in word recognition tasks. In bioinformatics, use for these models were found for example in gene prediction [13] and they are an integral part of many approaches to genome-wide nucleosome predictions.

Hidden Markov model consists of states interconnected by transitions. Let us call the state sequence the path, π. Then transitions between states are characterized by the following equation:

$$a_{kl} = P(\pi_i = l | \pi_{i-1} = k) \tag{3.1}$$

where π_i denotes the i-th state in the path.

To model the beginnning of the process we introduce a begin state. The transition probability of a_{0k} from this state to state k can be considered as the probability of starting in state k. We can treat similarly the probability of ending in state k by a_{k0}.

Every state is furthermore defined by emission probabilities, $e_k(b)$, probabilities that symbol b is seen when in state k.

$$e_k(b) = P(x_i = b | \pi_i = k) \tag{3.2}$$

HMM can be used as both a generator of sequences, and as a model for how a given sequence was generated. We will be interested mostly in the latter one, looking for the most probable path π^* through HMM for given sequence x.

$$\pi^* = \arg\max_{\pi} P(x, \pi) \tag{3.3}$$

This path can be found recursively. Suppose the probability $v_k(i)$ of the most probable path ending in state k with observation i is known for all the states k. Then these probabilities can be calculated for next observation x_{i+1} as:

$$v_l(i+1) = e_l(x_{i+1}) \max_k(v_k(i)a_{kl}) \tag{3.4}$$

All paths have to start in the begin state (label it 0), thus $v_0(0) = 1$. By keeping pointers backwards, the actual path can be found by backtracking. This procedure applied on sequence x of length L is illustrated below and is known as the Viterbi algorithm.

Initialization $(i = 0)$
$v_0(0) = 1$
$v_k(0) = 0$ for $k > 0$

Recursion $(i = 1, \ldots, L)$
$v_l(i) = e_l(x_i) \max_k(v_k(i-1)a_{kl})$
$ptr_i(l) = \arg\max_k(v_k(i-1)a_{kl})$

Termination
$P(x, \pi^*) = \max_k(v_k(L)a_{k0})$ $\pi^*_L = \arg\max_k(v_k(L)a_{k0})$

Backtracking $(i = L, \ldots, 1)$
$\pi^*_{i-1} = ptr_i(\pi^*_i)$

Apart from decoding sequences of observations, one need to be able to fit the model, derive emission and transition probabilities, to given training data. When all the paths are known, we can count the number of times each particular transition or emission is observed in the training set. Let these be A_{kl} and $E_k(b)$. Then the maximum likelihood estimators for a_{kl} and $e_k(b)$ are given as follows:

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \tag{3.5}$$

$$e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')} \tag{3.6}$$

If the path of states is unknown, then there exist two commonly used approaches — Viterbi training or Baum-Welch training technique.

The idea behind the Viterbi training is to compute the most probable path for the input sequence and then treat this path as if it was a previously known one, thus applying the same equations (3.5 and 3.6) to derive new parameters of the model.

The Baum-Welch algorithm is an iteration method that first estimates A_{kl} and $E_k(b)$ by considering probable paths for the training sequences using the current values of a_{kl} and $e_k(b)$. New values are derived in the same way as in previous techniques. This process is repeated until some stopping criterion is reached.

For further details about mentioned techniques, see [16] or [7].

3.2 Support vector machines

Support vector machines (or SVM) are supervised classification algorithm that separates two groups of data according ot given attributes. A training set is mapped onto a feature space and the algorithm looks for a hyperplane, which separates positive and negative examples maintaining a maximum margin from any point in the training set. The classification of an input data can be determined by mapping it into the same feature space and deriving the side of the separating plane on which the input lies.

3.2.1 Peckham et al.

The SVM algorithm was applied to nucleosome positioning problem in [15]. DNA fragments of length 50 base pairs labeled as nucleosome forming or inhibiting sequences, depending on a hybridization score obtained experimentally, where low score indicates nucleosome inhibiting fragment and high score otherwise, have been used as a dataset. The SVM classifier was trained on the 1000 strongest and 1000 weakest nucleosome forming fragments. Idea of this approach is to look at each sequence as on a fixed-length vector consisting of k-mer frequencies for $k = 1$ to 6 (A, T, C, G, AA and so on).

Peckham *et al.* identified GT/AT richness as the feature most responsible for distinguishing nucleosome forming or inhibiting sequences, which is consistent with previous findings that AT-rich intergenic regions are nucleosome-free. While GT/AT-richness of a sequence is the strongest factor among used k-mer frequencies, no individual k-mer could achieve results as good as the SVM relying on all k-mers.

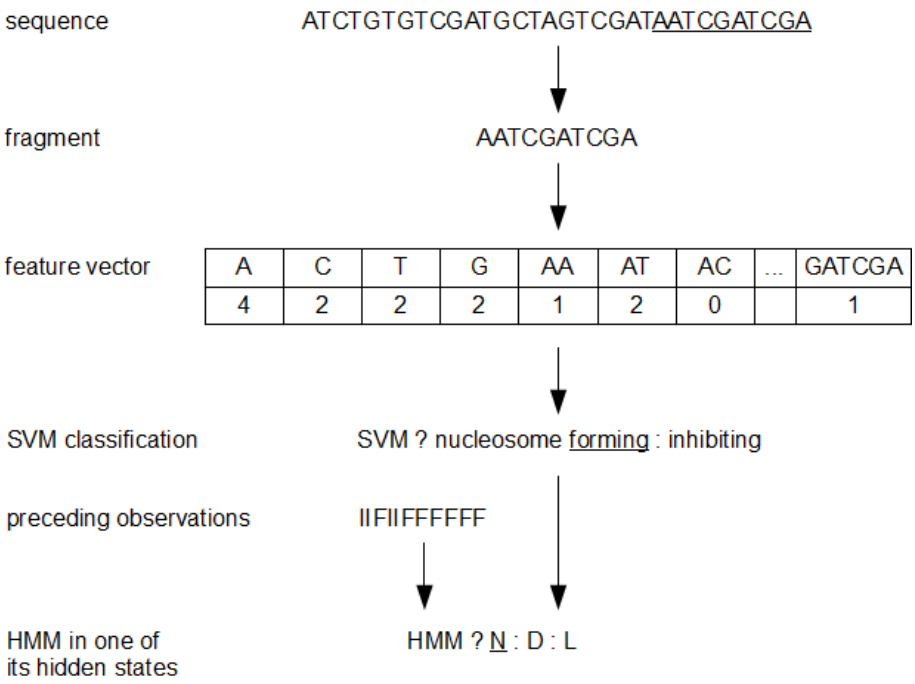


Figure 3.1: Schematics of simplified version of model described. Here we take fragments of 10bp with a 2bp step.

To predict nucleosome positions genome-wide, Hidden Markov Model was used to derive boundaries of predicted nucleosomes. Values obtained by the SVM classifier for 50bp

fragments of DNA serve as observed states of HMM, made of three hidden states - a well-positioned nucleosome (N), a delocalized nucleosome (D) and a linker region (L). The HMM uses both, the value from the SVM for an actual fragment as well as the probable state of DNA fragments preceding it, to determine its state. The HMM takes overlapping 50bp fragments of DNA with a step of 20bp as an input and results in six to eight probes indicating a well-positioned nucleosome and nine or more indicating delocalized nucleosome.

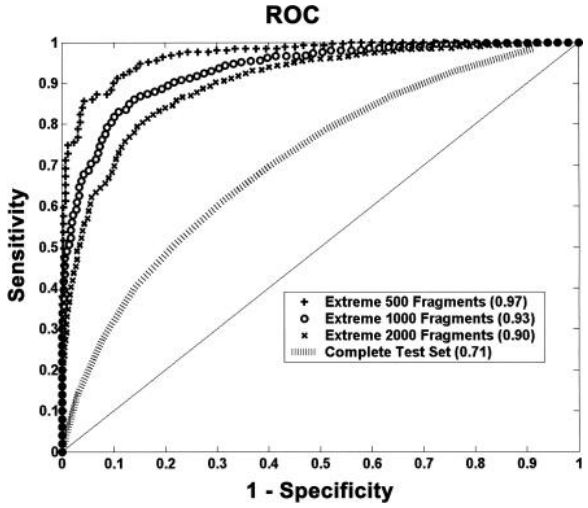


Figure 3.2: Performance of the SVM algorithm on classification of DNA fragments. Extreme fragments stand for fragments with the highest or lowest hybridization score. Source: [15].

The SVM itself achieved a ROC score of 0.71, when classifying only fragments contained in mentioned dataset. This score got higher, up to 0.97, by reducing the test set to fragments with extreme values of hybridization (performance measured by authors is shown on figure 3.2).

Genome-wide predictions successfully predicted up to 50% of well-positioned nucleosomes (with tolerance ±40bp) determined experimentally, compared to 33% expected by chance ([15]). This result is also in accord with studies [11] claiming that $\sim 50\%$ of nucleosomes are positioned by sequence.

3.3 Position specific scoring matrices

Fixed length of DNA wrapped around nucleosomes and positional preferences of specific dinucleotides makes Position specific scoring matrices (or PSSMs), and their various modifications, suitable for nucleosome prediction.

3.3.1 Kaplan et al.

Pioneering work on application of PSSMs to nucleosome prediction was done by Kaplan *et al.* in [11]. The model developed here is based on position-dependent features of nucleosome sequences and their 5-mer usage preferences. The model has two components, P_N representing the distribution over dinucleotides at each position along the nucleosome, and thus capturing the periodic signal of dinucleotides, and the second on, P_L, that serves as representation of the position independent distribution of nucleosomes over sequences

of length 5bp describing pentamers favoured or disfavoured by nucleosomes. This model assigns a score to every 147bp sequence S defined as 3.7.

$$Score(S) = \log \frac{P_N(S)}{P_L(S)} \tag{3.7}$$

$$P_N(S) = P_{N,1}(S[1]) * \prod_{i=2}^{147} P_{N,i}(S[i]|S[i-1]) \tag{3.8}$$

where $P_{N,i}$ stands for conditional probability distribution over nucleotides at position i given the nucleotide at the position $(i-1)$.

$$P_L(S) = P_1(S[1]) \prod_{i=2}^{147} P_1(S[i]|S[\max(1, i-4), \ldots, S[i-1]) \tag{3.9}$$

with P_1 being the position independent component of P_L.

P_N component was derived from aligned nucleosome-bound sequences, with each sequence from the entry dataset included twice — in its original and reverse complement form. For each position i was then calculated a dinucleotide distribution $P_{N,i}$ estimated form dinucleotide counts at positions $[i-2, i-1]$, $[i-1, i]$ and $[i, i+1]$. Combining dinucleotide counts at neighbouring positions is mainly motivated by an experimental evidence that small 1bp changes in spacing of key nucleosome DNA motifs can occur with relatively small cost to the free energy of histone-DNA interactions. Finally, resulting distributions were normalized to remove sequence composition biases.

P_L component serves as representation of generally favoured or disfavoured sequences within nucleosomes. P_1 defines a probability distribution over all 1024 possible pentamers in such way that higher probabilities correspond to disfavoured sequences. We can look at this component as a Markov model, which includes contributions both from nucleosome favoured and disfavoured sequences, with disfavoured sequences having relatively high probability and favoured ones otherwise.

In the end, the performance of the second (P_L) component alone is nearly as good as that of the full model, while the P_N proved to be highly predictive, but slightly worse than complete or P_L model, which may suffice for practical purposes of prediction.

This tool is available online at http://genie.weizmann.ac.il/software/nucleo_prediction.html.

3.3.2 Position-correlation scoring function

Another work dealing with PSSMs is [25], where a method called position-correlation scoring function was developed. Authors of this paper have focused more on linker DNA preferences, especially tetramer usage, which they found more distinctive then that of dinucleotides or any other k-mer with $k < 6$.

The classification model is based on extension of PSSMs called position-correlation scoring function. This function assigns to each position of PSSM weight representing the amount of information contained within given position. Function is described by parameter $M_k(i)$ defined as 3.10.

$$M_k(i) = \sum_{j=1}^{4^k} \frac{(\frac{f_i(j)}{N} - \frac{1}{4^k})^2}{\frac{1}{4^k}} \tag{3.10}$$

where $f_i(j)$ is the real count of the j-th element of k-mer at the position i along sequences and N is the number of sequences. This $M_k(i)$ parameter accounts for the deviation degree of any k-mer frequency from random distribution at the i-th position along sequences. The larger the value is, the stronger sequence bias at the i-th site is.

Conventional PSSM is used to reflect probability of oligonucleotides observed at position i 3.11.

$$P_i(j) = \frac{f_i(j) + s(j)}{N + S} \tag{3.11}$$

with $s(j)$ and S as pseudocounts added to eliminate null values before the log conversion. These probabilities are afterwards transformed into log space for easier computations and weights of positional weight matrix (log version of PSSM) for nucleosomal occupancy are then determined as in 3.12.

$$W_i(j) = \ln \frac{P_i(j)}{p_0} \tag{3.12}$$

p_0 being the expected background probability of each element of sequence.

Merging together positional weight matrix (PWM) and position-correlation parameter M_k, we come to position-correlation scoring function (PCSF) defined for sequence of L base pairs as 3.13.

$$S_1 = \frac{\sum_{i=1}^{L}(M_k(i) * W_i(j) - M_k(i) \min W_i(j))}{\sum_{i=1}^{L}(M_k(i) \max W_i(j) - M_k(i) \min W_i(j))} \tag{3.13}$$

Using this formula, we end up with PWM broaden by an additional weight for every position, which presents informational value of given position. S_1 spreads from 0 to 1, the higher the value is, the higher the nucleosome occupancy is, if trained on nucleosomal DNA data.

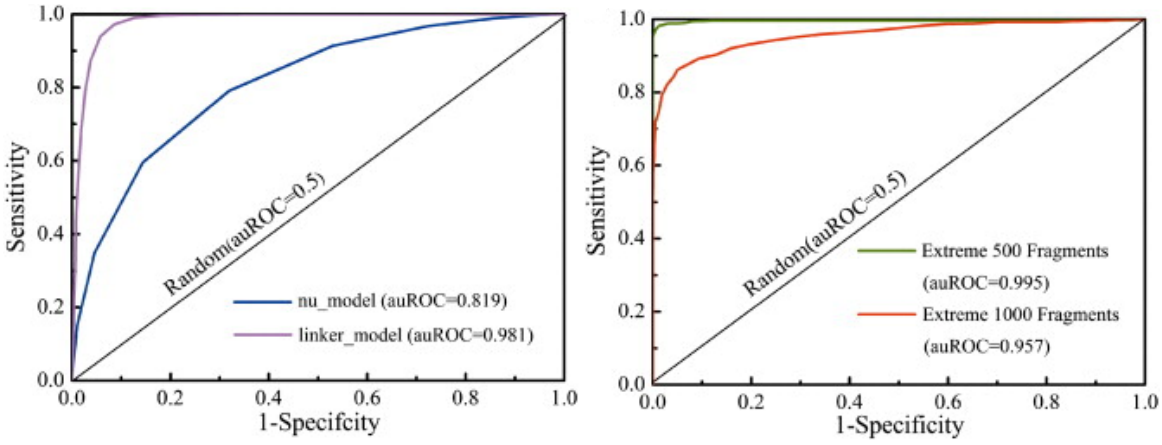
The algorithm computes this PCSF for each position in a sequence moving the sliding window of length 150bp with a step of 1bp. Such PCSF profile is then smoothed by sliding average of a window with the same length, when smoothed values are taken as nucleosome occupancy potential at the central position of this window.

Training this model on both nucleosome and linker DNA sequences had shown that by training on linker sequences datset, one can achieve significantly better results, then the other way around (3.3a). PCSF slightly outperforms algorithm presented in section 3.2.1 dedicated to SVM.

3.3.3 Markov chains within duration HMM

NuPop tool [24], available online at http://nucleosome.stats.northwestern.edu/, introduce modeling of the linker DNA distribution to the nucleosome positioning prediction. It is based on a modification of classic Hidden Markov models, called duration HMM.

Model, HMM, consists of two states, 147bp fixed-length state N (nucleosome) and L state modeling variable length of linker DNA. At the end of each state, model makes transition to another one. For nucleosome state, 4th order time-dependent Markov chain was trained, which is in other words a higher order PSSM. Also 4th order, but homogenous, Markov chain was trained for linker DNA. These chains should distinguish k-mer usage preferences between nucleosome and linker DNA.



(a) (b)

Figure 3.3: (a) Graph shows the difference in performance of the model due to used training set. Blue liner represents ROC of model trained on nucleosome sequences, violet one on linker DNA. (b) Performance of the model in the extreme fragments from [15]. PCSF performs slightly better then SVM by Peckham *et al.*. Source: [25]

Probability of observing sequence e of length 147bp as nucleosome is computed simply as a product of probabilities for both strands under the 4th Markov chain model of state N. On the other hand, observed linker DNA sequence e of length k carries two pieces of information, the length and emitted letters. Probability of observing sequence e is then defined by equation 3.14.

$$P_L(e) = G_L(e|k)F_L(k) \tag{3.14}$$

where $F_L(k)$ denotes the linker DNA length distribution for given species (this distribution differs from species to species) and $G_L(e|k)$ stands for the probability emitted by homogenous Markov chain of state L again including both strands.

Suppose x is a genomic DNA sequence of length n and z is a corresponding hidden state path, where $z_i = 1$ if x_i is covered by nucleosome state and 0 otherwise. Suppose that path z partitions x into k consecutive nucleosome or linker state blocks, where nucleosome blocks have a fixed length 147bp, whereas the length of linker blocks vary. Mark these blocks as $y = y_1, \ldots, y_k$ and their state identification as $s = s_1, \ldots, s_k$. The probability of observing (x, z) is then given by formula 3.15.

$$P(x, z) = \pi_0(s_1)\pi_e(s_k)\prod_{i=1}^{k}\{P_N(y_i)\}^{l(s_i=1}\{P_L(y_i)\}^{l(s_i=0)} \tag{3.15}$$

where $\pi_0(s_1)$ and $\pi_e(s_k)$ are probabilities that the chain initializes and ends with the state s_1 and s_k respectively, and l is an indicator function. Since authors assume that chromatin sequence must start with and end in a linker state, we can replace $\pi_0(s_1)\pi_e(s_k)$ by 1. The most probable path of hidden states z can be then obtained traditionally using Viterbi algorithm.

3.3.4 Bendability matrix

In [8], authors derived a consensus sequence for dinucleotides within nucleosomal 10bp repeats. It is called bendability matrix and it represents deformational affinity of given dinucleotide at every position of the period.

In other words, it is typical PSSM for 10bp repeats that nucleosomes sequences should ideally consist of, created after several approximations, to fit onto periodic repeats within 147 base core sequences. It starts and ends with the same dinucleotide (and thus has 11 columns). Tool based on this matrix is available online at http://www.cs.bgu.ac.il/~nucleom/.

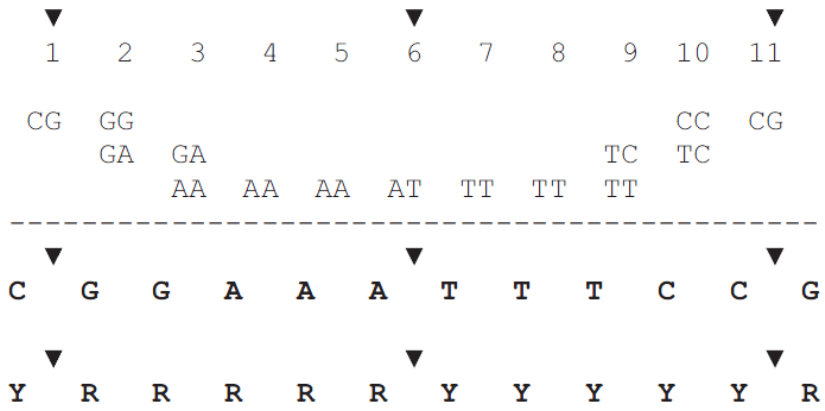


Figure 3.4: The consensus sequence of highest positional affinity dinucleotides within 10 base period. Triangles show symmetry axes. Y stands for pyrimidine base, R for purine one. Source: [8]

3.4 Other approaches

This section completes the list of examined methods with three other approaches, giving only brief information about first two of them and explaining in more detail the last one, based on different principles than approaches explained above.

In speech signal processing, Fourier or Wavelet analysis are performed in order to obtain periodic characteristics of explored signal. The latter one is part of the N-score method [27], which transforms sequences of 130bp into 16 130-dimensional vectors, one vector for each dinucleotide. Every three positions are afterwards averaged resulting into 128-dimensional vectors, on which wavelet analysis is applied to discover desired periodicities. Finally, a logistic regression model classifies given sequences as DNA occupied by nucleosomes or not, according to computed N-score. This procedure is incorporated into Hidden Markov model for genome-wide predictions.

NuScore [20] is an online tool (available at http://compbio.med.harvard.edu/nuScore/) for nucleosome positioning prediction based on the estimation of energy cost of the structural deformation imposed on DNA in nucleosome core.

3.4.1 Mirror position filtering

Another method, based solely on the periodic occurrence of dinucleotides, was proposed in hypothesis [23]. Interesting mainly because it relies only on prior knowledge and thus does not require any training data.

Let us define two sets, B_1 containing four nucleotide bases and B_2 composed of 10 unique dinucleotides (reverse complement dinucleotides are considered equivalent). Assume that DNA sequence is represented by discrete function $x(n)$, where $x(n) \in B_1$ and $n = 0, 1, 2, \ldots, N$ with N being a length of sequence. For each dinucleotide from B_2, a following delta function is defined to represent the positions of nucleotide bases:

$$x_b(n) = \ldots + \delta(\tilde{n}\ n_{kl}) + \delta(\tilde{n}\ n_k) + \delta(\tilde{n}\ n_{kr}) + \ldots \tag{3.16}$$

where $\delta(\tilde{n}\ n_k)$ is an impulse represented by the Dirac delta function indicating presence of b at position nk. For example, let us have $x(n) = \{GACTAGCACGGTAC\}$, then $x_{AC}(n) = \delta(\tilde{n}\ 1) + \delta(\tilde{n}\ 7) + \delta(\tilde{n}\ 12)$.

To produce a $\sim$ 10.5bp periodicity, we would expect an impulse $\delta(\tilde{n}\ n_k)$ for each position n_k having an impulse $\delta(\tilde{n}\ n_{kl})$ on the left-hand side 10 to 11bp away as well as $\delta(\tilde{n}\ n_{kr})$ on the right-hand side with the same distance. Defining $d_L(b, n_k)$ and $d_R(b, n_k)$ as a distance of $\delta(\tilde{n}\ n_{kr})$ to the impulse closest to the position 10.5bp away to the left and to the right respectively, we should end up with $d_L(b, n_k) = 10$ or 11 and $d_R(b, n_k) = 10$ or 11 as many times as possible in nucleosomes and no such impulses in a linker DNA. Real values of $d_L(b, n_k)$ and $d_R(b, n_k)$ may deviate from the ideal ones and that is why a mathcing function $f(d)$ is used to measure the contributions of $\delta(\tilde{n}\ n_{kl}), \delta(\tilde{n}\ n_k)$ and $\delta(\tilde{n}\ n_{kr})$ to the periodicity. This function returns large values for d close to ideal 10.5 and decrease as d moves away from this value.

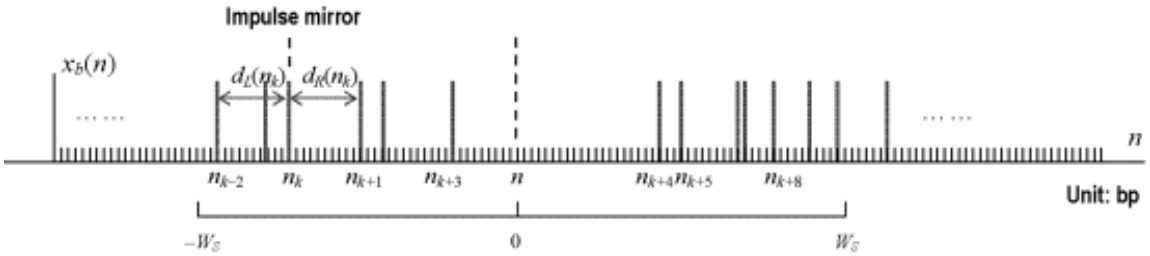


Figure 3.5: Occurrences of dinucleotide along DNA sequence. A dinucleotide should have ideally two mirror images, 10 to 11bp to the left and to the right, to produce the periodicity of $\sim$ 10.5bp in a nucleosome. Source: [23]

To detect nucleosomes along the DNA sequence, a sliding window with the size of $2W_s+1$ (as nucleosomes wrap DNA of 147bp, $W_s = 73$) is used to accumulate the contributions from all dinucleotides within the window and if the score $S(n)$ of the window is over a defined threshold, then the window's position is marked as nucleosome.

$$S(n) = \sum_{b \in B_2} \sum_{n - W_s \leq n_k \leq n + W_s} f(d_L(b, n_k)) + f(d_R(b, n_k)) \tag{3.17}$$

Despite its simplicity, authors claim that almost 50% of predicted nucleosomes were real nucleosomes within the yeast genome.

Chapter 4

Implementation of nucleosome prediction method

Implementation and all experiments were done in programming language Python in version 2.7 (http://www.python.org) using freely available libraries:

`scipy` (http://scipy.org)
: for most of mathematical operations

`matplotlib` (http://matplotlib.org)
: for graphical outputs (figure plotting etc.)

`biopython` (http://biopython.org)
: for manipulation with sequence data

Python interpret and these packages are required to run any scripts included with this thesis.

Application and other scripts are available together with other supplementary material at http://bioware.fit.vutbr.cz/mediawiki/index.php/Nucleosome_prediction.

4.1 Used terminology

To ensure that reader completely understands ideas presented on the following pages, it might be useful to clarify frequently used terms occuring throughout the rest of this thesis.

Nucleosome sequence
: part of DNA sequence occupied by nucleosome (wrapped around the histone core)

Centre of nucleosome
: position in given DNA sequence around which is particular nucleosome centered or in other words 74th nucleotide within a sequence occupied by given nucleosome (nucleosome sequence)

Linker sequence or linker
: DNA sequence lying in-between adjacent nucleosome sequences

4.1.1 Statistics for evaluation of performance quality

Following metrics and definitions are used for evaluation of examined approaches. Predicted nucleosome is considered a true positive match (TP), if distance between predicted centre and centre of nucleosome within given dataset lies within specified tolerance ($\pm$35bp if not stated otherwise — for instance in [15, 24]). Predictions, which does not satisfy above condition are regarded as false positive matches (FP). We define false positives (FP) as entries from dataset that are located outside boundaries determined by tolerance around predicted centres of nucleosomes.

Furthermore, *sensitivity*, which gives the fraction of nucleosomes from dataset that are predicted, is defined as usually (4.1).

$$sensitivity = \frac{TP}{TP + FN} \tag{4.1}$$

Because we are not able to define true negative matches conveniently, we use alternative formula (4.2) for *specificity*, called also *positive predictive value* (for example [21]), which gives the fraction of predicted nucleosomes that are in the dataset.

$$specificity = \frac{TP}{TP + FP} \tag{4.2}$$

4.2 Dataset

In this project, a map of nucleosome positions in yeast published in [4] has been used as primary dataset. Map defines positions of nucleosome centres throughout the whole yeast genome, specifically for UCSC-SAC2 assembly, and consists of two parts — redundant map (`http://www.nature.com/nature/journal/vaop/ncurrent/extref/nature11142-s3.txt`) allowing 351, 264 nucleosomes to overlap arbitrarily and unique map (`http://www.nature.com/nature/journal/vaop/ncurrent/extref/nature11142-s2.txt`) containing 67, 543 nucleosomes, where two neighbouring nucleosomes are allowed to overlap by no more than 40 base pairs and which is a subset of redundant map.

Besides positions of nucleosome centres each entry of mentioned maps include *nucleosome centre positioning score* (*NCP*) and *NCP score-to-noise ratio* values. The *NCP* score provides a measure of the relative amount of nucleosomes centered at given position and its score-to-noise ratio, which is the *NCP* score adjusted with respect to average noise at the same position. More detailed description of these measures and methods for obtaining them can be found in supplementary material of [4].

4.2.1 Analysis of dataset

In the beginning, we will take a closer look at well-known features of nucleosome positioning in our dataset and to what extent are these features represented here.

Linker length distribution

We mentioned previously (section 2.1.2) that linkers emits non-random distribution of their lengths, which differs across species. In our dataset, linker lengths are distributed with mean 48.2bp and median 26bp (also in accord with [2]). Histogram below (figure 4.1) shows three strong peaks at lengths 3, 15 and 24bp and somewhat weaker peaks at 34, 44 and 54bp.

These peaks are approximately 10bp apart, what could be related to the length of DNA helix turn ($\sim$ 10.5bp) [4].

Apart from these linkers, there are 16,932 nucleosome overlaps that results in no linker DNA in-between adjacent nucleosomes.

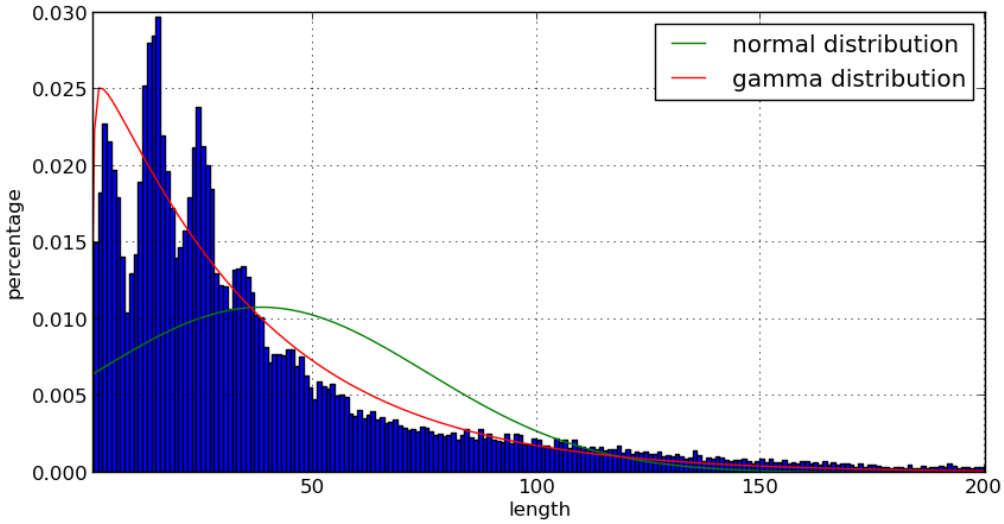


Figure 4.1: Histogram of linker lengths shorter than 200bp (which represents 97% of all linkers), based on the unique set of nucleosomes, and normal and gamma distributions fitted to these lengths.

Differences between composition of nucleosome and linker sequences

Many computational methods of nucleosome prediction rely on differences in k-mer frequencies in nucleosome and linker DNA. Especially k-mers made exclusively of A and T are considered nucleosome inhibiting elements of DNA (more on page 26).

Linker		Nucleosome		Genome-wide	
dimer	%	dimer	%	dimer	%
AA(TT)	0.1078	AA(TT)	0.1083	AA(TT)	0.1080
AT	0.0859	AT	0.0903	AT	0.0894
TA	0.0708	TA	0.0738	TA	0.0733
TG(CA)	0.0635	TG(CA)	0.0651	TG(CA)	0.0648
TC(GA)	0.0618	TC(GA)	0.0624	TC(GA)	0.0623
AG(CT)	0.0576	AG(CT)	0.0585	AG(CT)	0.0583
AC(GT)	0.0522	AC(GT)	0.0527	AC(GT)	0.0527
CC(GG)	0.0408	CC(GG)	0.0383	CC(GG)	0.0389
GC	0.0408	GC	0.0365	GC	0.0374
CG	0.0341	CG	0.0281	CG	0.0293

Table 4.1: Table showing frequency of dimers in linker, nucleosome and genome sequence.

Variations in dimers and trimers frequencies within linker and nucleosome sequences

are shown in table 4.1 and 4.2. These values were obtained by counting all occurrences of particular k-mers in nucleosome and linker sequences on both strands, that is why reversed complementary k-mers have the same frequency. Although the order (based on frequency) of most k-mers does not differ, one can observe decreased frequency of AA or AAA words in nucleosome sequences compared to linker or genome-wide distribution. A bit surprisingly, words made of Gs and Cs are also repeated more in linker sequences, while we would expect it other way around, because G+C content is also considered as one of possible nucleosome positions determinant [19].

Linker		Nucleosome		Genome-wide	
trimer	%	trimer	%	trimer	%
AAA(TTT)	0.0409	AAA(TTT)	0.0388	AAA(TTT)	0.0391
AAT(ATT)	0.0278	AAT(ATT)	0.0295	AAT(ATT)	0.0291
GAA(TTC)	0.0236	TAT(ATA)	0.0245	TAT(ATA)	0.0242
TAT(ATA)	0.0231	GAA(TTC)	0.0238	GAA(TTC)	0.0237
CAA(TTG)	0.0220	CAA(TTG)	0.0234	CAA(TTG)	0.0231
AAG(CTT)	0.0214	TTA(TAA)	0.0221	TTA(TAA)	0.0219
TTA(TAA)	0.0213	AAG(CTT)	0.0217	AAG(CTT)	0.0216
TCT(AGA)	0.0196	TCT(AGA)	0.0204	TCT(AGA)	0.0203
TGA(TCA)	0.0194	TGA(TCA)	0.0203	TGA(TCA)	0.0202
ATG(CAT)	0.0179	ATG(CAT)	0.0184	ATG(CAT)	0.0183
AAC(GTT)	0.0175	AAC(GTT)	0.0182	AAC(GTT)	0.0180
ATC(GAT)	0.0168	ATC(GAT)	0.0177	ATC(GAT)	0.0176
ACA(TGT)	0.0165	ACA(TGT)	0.0173	ACA(TGT)	0.0172
ACT(AGT)	0.0146	ACT(AGT)	0.0152	ACT(AGT)	0.0151
TGG(CCA)	0.0145	TGG(CCA)	0.0150	TGG(CCA)	0.0149
TAC(GTA)	0.0139	TAC(GTA)	0.0142	TAC(GTA)	0.0142
TCC(GGA)	0.0129	TAG(CTA)	0.0129	TAG(CTA)	0.0128
GCA(TGC)	0.0128	TCC(GGA)	0.0127	TCC(GGA)	0.0127
TAG(CTA)	0.0125	CAG(CTG)	0.0126	CAG(CTG)	0.0126
CAG(CTG)	0.0124	GCA(TGC)	0.0123	GCA(TGC)	0.0124
AGC(GCT)	0.0118	ACC(GGT)	0.0115	ACC(GGT)	0.0116
ACC(GGT)	0.0116	AGC(GCT)	0.0114	AGC(GCT)	0.0115
CCT(AGG)	0.0115	CCT(AGG)	0.0113	CCT(AGG)	0.0114
GAG(CTC)	0.0114	GAG(CTC)	0.0111	GAG(CTC)	0.0112
CAC(GTG)	0.0111	CAC(GTG)	0.0105	CAC(GTG)	0.0106
CGA(TCG)	0.0099	GAC(GTC)	0.0097	GAC(GTC)	0.0097
GAC(GTC)	0.0098	CGA(TCG)	0.0089	CGA(TCG)	0.0091
CGT(ACG)	0.0095	CGT(ACG)	0.0085	CGT(ACG)	0.0087
GGC(GCC)	0.0088	GGC(GCC)	0.0076	GGC(GCC)	0.0078
GGG(CCC)	0.0075	GGG(CCC)	0.0065	GGG(CCC)	0.0067
CGC(GCG)	0.0074	CGG(CCG)	0.0054	CGG(CCG)	0.0058
CGG(CCG)	0.0072	CGC(GCG)	0.0051	CGC(GCG)	0.0056

Table 4.2: Table showing frequency of trimers in linker, nucleosome and genome sequence.

It is worth mentioning that we also compared word composition of longer and shorter linker sequences, setting boundary between these two classes around the length of 60bp,

but we did not observe any significant differences, when the most notable deviation was the higher portion of polyA:T k-mers within longer linker DNA, which can be explained by their higher occurrence genome-wide.

Nucleosome directionality

Another step in our analysis was creation of PSSM, as they are used extensively in existing methods and they also capture periodic pattern of repeating dinucleotides. Mononucleotide PSSM is shown in figure 4.2 and one can make following observations:

- Periodic pattern of dinucleotides composed by A and T in phase of 5bp with dinucleotides made of C and G is clearly visible.
- A at position -3 and T at position +3 are highly overrepresented ([4])
- PSSM is almost completely symmetrical in its complement

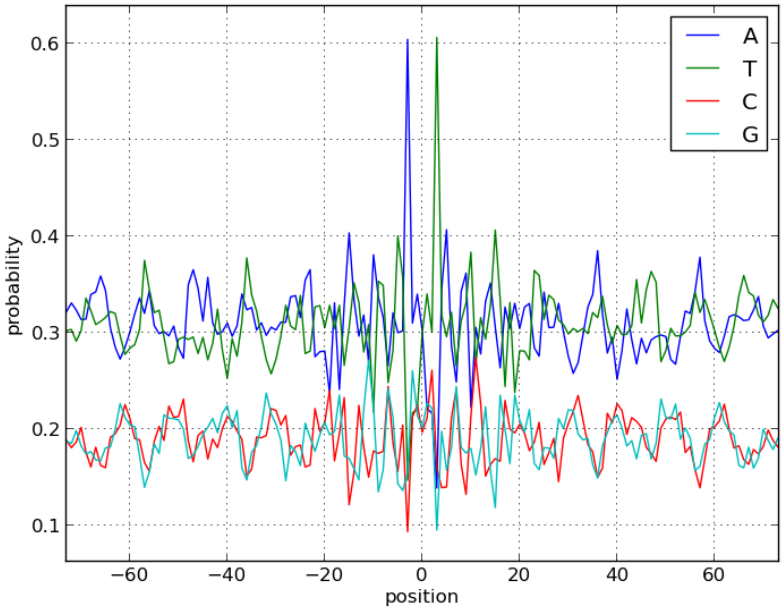


Figure 4.2: Position specific scoring matrix derived from unique set of nucleosomes centered around nucleosome dyad.

Periodic pattern is well-known feature of nucleosome sequences, more on A and T at positions -3 and +3 can reader find in [4], so we will examine closely the symmetry of this matrix.

This symmetry is represented by higher frequency of adenine at the 5'end of nucleosome sequence, which decreases towards the 3'end, and the same applies for thymine, but in the other direction. Also peaks in frequency of particular nucleotides on the one half of the nucleosome sequence result in the same peak of its complement with the same distance to the nucleosome centre on the other half of the sequence. We can make similar proposition

concerning peaks in frequencies about cytosine and guanine. Interestingly, these complementary peaks are also visible in dinucleotide PSSM (figure 4.6).

These findings are in accordance with [17], where authors also identified the highest local density of Gs and the lowest of Ts occurring ~ 40 nucleotides upstream of the centre, As and Cs likewise. This might be related to the close proximity of the two superhelical coils within nucleosomes, where regions ~ 80 nucleotides apart are brought close to each other due to the DNA winding around the core. If this close proximity affects nucleosome sequence preference, one could recognize some motif in this ~ 40 region.

Let us assume that nucleosomes have something one could call direction — positive and negative. We declare nucleosome positively oriented if it fulfills following requirements:

1. Number of adenines at the 5'end of nucleosome sequence is greater than that at the 3'end

2. Number of thymines at the 5'end of nucleosome sequence is lower than that at the 3'end

3. Score of PSSM is larger than score of its reversed version (thanks to its symmetry, we just replace bases with their complements)

Rules for negative orientation are defined likewise. These conditions are quiet strict and leave more than half of the nucleosomes without a direction. To assign directions to these nucleosomes, we omitted conditions one by one. Numbers of nucleosomes satisfying directionality conditions are shown in table 4.3.

Applied conditions	Positive orientation	Negative orientation	Total
1. and 2. and 3.	25,667 (38%)	3,304 (5%)	28,971 (43%)
1. and 2.	26,901 (40%)	16,899 (25%)	43,800 (65%)
1. or 2.	46,780 (69%)	36,104 (53%)	67,543 (100%)
3.	60,692 (90%)	6,842 (10%)	67,534 (> 99%)

Table 4.3: Amount of nucleosomes fulfilling specified conditions of directionality.

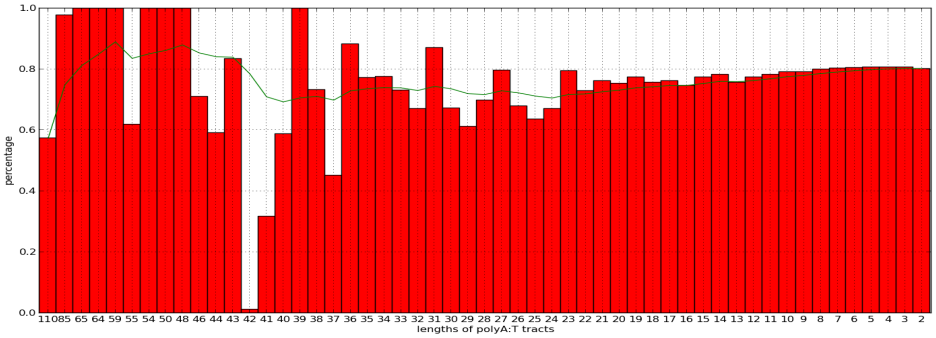
Although we did not pursue any extensive analysis of nucleosome orientation related to functional sites of DNA, these negatively oriented nucleosomes might have something to do with replication origins, as almost every one on chrI has some negatively oriented nucleosomes nearby (e.g. ARS103 is almost completely covered by negative ones), but it might be also just coincidence. Besides, we did not observe any significant clustering of negatively oriented nucleosomes.

Note on polyA:T tracts

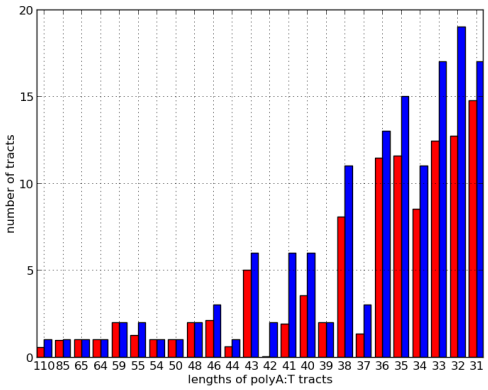
Most literature (e.g. [2], [9], [12]) describes polyA:T tracts as nucleosome inhibiting elements of DNA sequence. These tracts occur mainly in promoter regions that have to be nucleosome depleted, otherwise transcription factor binding sites would be inaccessible to transcription factors in these regions.

Our observations slightly differ. We looked at stretches of DNA sequence consisting exclusively of A, T or both of them. Results are shown on 4.3 below, and while tracts made of solely A (4.3c) and T (4.3d) follow this assumption in general (although there are

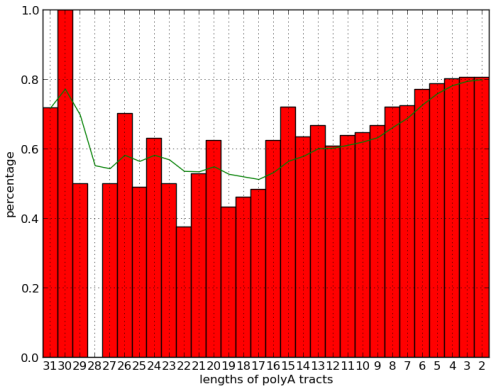
some exceptions), tracts composed by both A and T (4.3a and 4.3b) tend to be occupied by nucleosomes more than expected.



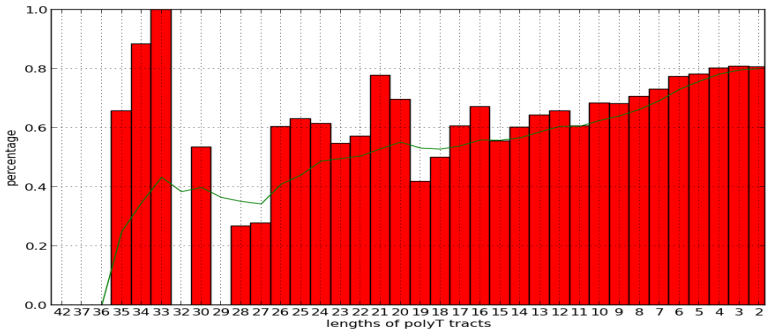
(a) PolyA:T tracts



(b) Histogram of longest polyA:T tracts



(c) PolyA tracts



(d) PolyT tracts

Figure 4.3: Coverage of polyA:T tracts by nucleosomes. Red bars represent percentage (or ratio in 4.3b) of tracts with particular length occupied by nucleosomes. Green line stands for cumulative percentage of tracts occupied by nucleosomes (defined for tracts of length i as $\frac{\sum_{j=i}^{maxlength} n(j)}{\sum_{j=i}^{maxlength} s(j)j}$, where $n(j)$ is total number of base pairs occupied by nucleosomes in tracts of length j and $s(j)$ is number of tracts of length j). Blue bars in 4.3b represent total amount of base pairs that tracts of particular length consists of.

Conclusion

In the end, we can summarize our initial analysis into following points related to our purposes:

- Nucleosome and linker sequences differ in frequencies of various k-mers, although G+C content might not play such important role in nucleosome positioning
- Lengths of linker sequences are not distributed randomly, but peaks at 3, 15 and 24bp
- Symmetry of PSSM in its complement means that it should not be necessary to look at both strands of DNA for making predictions of nucleosome positions
- polyA:T tracts might not be absolutely reliable signal of nucleosome depletion

4.3 Evaluation of nucleosome positioning features

In chapter 3, we have identified two main features of nucleosome sequences useful for their prediction — k-mer composition and periodic occurence of particular dinucleotides. On following pages, we will deal with these features trying to obtain their „predictive power“ throughout our dataset.

We describe features by specific scoring function, which is then applied on sequence within sliding window moved along chromosome I, transforming nucleotide sequence into series of numerical values. Peaks of this profile (or series), extracted by method commonly used for mass spectrum analysis [6], are marked as nucleosome predictions assuming that the representation of these features is highest around nucleosome centers.

4.3.1 Peak detection or predictions without linker length information

Numerical profiles generated by examined scoring functions kept variable levels of values throughout chromosome I. For instance, there are long regions where log-ratios (next section 4.3.2) yielded very low values, even though they were occupied by nucleosomes. In these places, profiles tend to peak, creating local maxima that can be far below global ones. That is why we choose peak detection method rather than use of some threshold.

Peak detection method [6] was chosen according to recommendations from [26]. It is part of `bioconductor` package (`http://bioconductor.org`) for programming language R and also part of the latest version of `scipy` library for python. Without going into much detail, the method first smooths input profile with continuous wavelet transform (CWT, in this case Mexican Hat wavelet) that also removes baseline automatically and then determines peaks in regard to signal to noise ratio and ridge lines, both obtained from CWT.

Code below demonstrates application of this CWT peak detection method.

```
source("http://bioconductor.org/biocLite.R")
biocLite("MassSpecWavelet")
values <- scan(input)
peaks <- peakDetectionCWT(values, SNR.Th = 6, nearbyPeak = FALSE)
write(peaks$majorPeakInfo$peakIndex, output)
```

This code is also part of `peaks.R` script (enclosed with other scripts), which can be used to extract peaks from profile stored in file named `input_file` and save them into `output_file` by running the following command in bash:

```
R --no-save --args input_file output_file < peaks.R
```

4.3.2 Log-ratio scoring function

We showed that frequencies of word composition differ between nucleosome and linker sequences 4.2.1. To interpret these variations, we introduce log-ratio scoring function (also in [11]).

Log-ratio $r(x)$ for k-mer x is defined as

$$r(x) = \log_2 \frac{p_{nucleosome}(x)}{p_{linker}(x)} \tag{4.3}$$

where $p_{nucleosome}(x)$ is a probability of k-mer x occuring in nucleosome sequence and $p_{linker}(x)$ in linker sequence respectively. We derive our log-ratios based on word composition of $5,000$ nucleosomes with the highest NCP score-to-noise ratios and randomly chosen $15,000$ linker sequences.

dimer	log-ratio
AT	0.0719
TA	0.0594
TG(CA)	0.0356
AG(CT)	0.0209
TC(GA)	0.0131
AC(GT)	0.0124
AA(TT)	0.0071
CC(GG)	-0.0913
GC	-0.1604
CG	-0.2792

Table 4.4: Table of log-ratios derived for dimers from all nucleosomes and linkers in dataset. Log-ratios used in our evaluations differs a bit as they are derived from $5,000$ best positioned nucleosomes and randomly chosen $15,000$ linker sequences.

A sequence is transformed into numerical profile by sliding window of length w_{length}, specifying log-ratio score at given position i as follows

$$score_i = \sum_{j=i-\frac{w_{length}-1}{2}}^{i+\frac{w_{length}-1}{2}} r(x_j) \tag{4.4}$$

where x_j is k-mer x with last letter at position j and the length of window being an odd number.

First, we examine to what extent experimentally obtained nucleosome occupancy [4] corresponds to our log-ratios as similar log-ratio function was used to obtain nucleosome

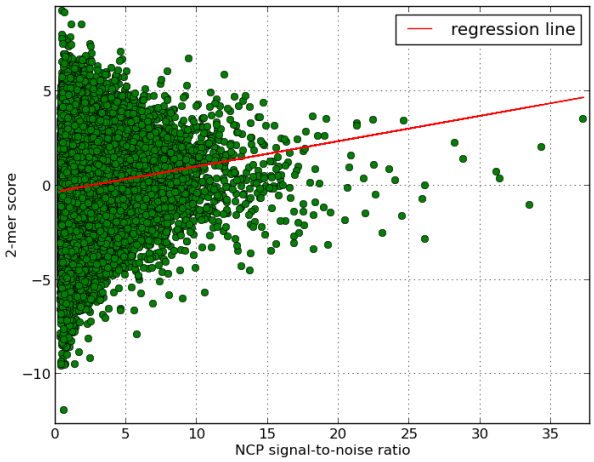
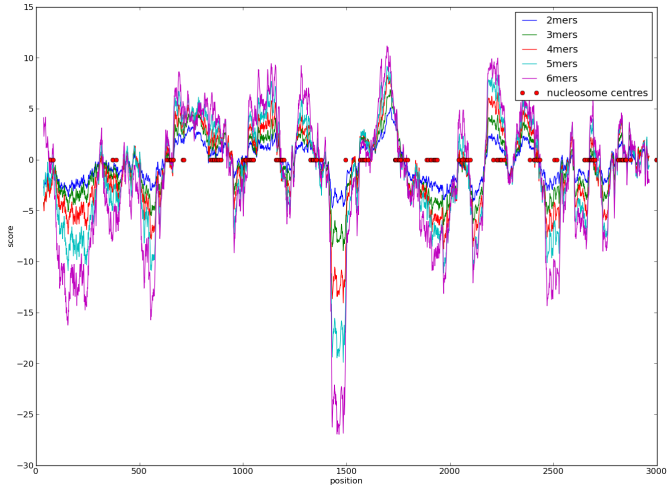
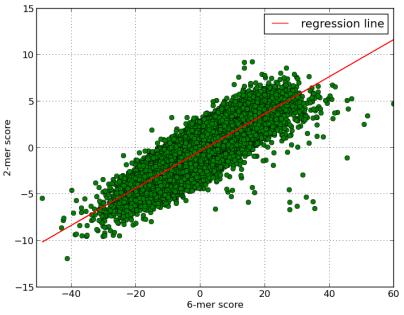


Figure 4.4: Correlation between NCP score-to-noise ratio on x-axis and log-ratio scoring function on y-axis. Regression line was fitted to the data to show weak positive correlation.

occupancy along DNA sequence in [11]. For this purpose, a window of 147bp is placed on center of all nucleosomes in our dataset and each nucleosome is scored in accord with equation 4.4. The results, counting only on dimer preferences, are depicted in figure 4.4. One can see that the variation of log-ratio scores decreases with higher NCP score-to-noise ratio values and that values are, although weakly, positively correlated (pearson's correlation coefficient 0.1055). We observe that nucleosomes with log-ratio score significantly lower than NCP score-to-noise ratio are mostly located in telomeric regions, which are made of long repeats of the same sequence and thus underlying sequence contains only minimum information.



(a) Log-ratio profiles of first 3000bp on chromosome I based on different length of k-mers.



(b) Correlation between log-ratio scores of nucleosomes based on dimers and hexamers.

Figure 4.5: Correlation of log-ratio scores.

Next, we predict nucleosome positions by applying peak detection method mentioned above (section 4.3.1) on profile created by sliding the window of length 74bp with 1bp step. Originally, length of the sliding window was set to 147bp, but experiments with this length proved that shorter window performs better. Relying sequentially on log-ratios, based solely on k-mers of specified length, revealed that profiles of different lengths of k-mers maintain almost the same shape (figure 4.5a). This outcome is somewhat to be expected, because longer k-mers are derived from the shorter ones, and results in only minor differences in performance between usage of dimer's and any other k-mers up to $k = 6$. Log-ratio scores of nucleosomes from our dataset derived from dimers and other longer k-mers are also highly correlated (figure 4.5b), yielding pearson's correlation coefficients near 1 (table 4.5).

k	2	3	4	5	6
coefficient	1	0.98	0.95	0.90	0.83

Table 4.5: Pearson's correlation coefficients showing correlation between log-ratio scores of nucleosomes based on dimers and other k-mers.

In the end, log-ratio scoring function peaks at more than a half of nucleosomes on chromosome I (the best results achieved are shown in table 4.6), which is in accordance with conclusions from [10] that sequence composition is responsible for positioning of approximately 50% of nucleosomes. Relatively large number of false positives is due to exclusion of any information concerning spacing between adjacent predictions, which is to be solved in next chapters. We attempted to decrease the amount of false positives by omitting peaks with score below given threshold, but it resulted in similar reduction of true positives.

Tolerance	True positives	False positives	False negatives	Sensitivity	Specificity
$\pm 10bp$	247	1,511	1,046	0.19	0.14
$\pm 20bp$	449	1,308	844	0.35	0.25
$\pm 35bp$	741	1,031	552	0.57	0.42

Table 4.6: Performance of log-ratio scoring function based on 6-mer preferences on chromosome I using 74bp long sliding window.

4.3.3 Fourier transform

Another key feature of nucleosome sequences is $\sim$ 10.5bp periodicity of dinucleotides AA/TT/AT/TA. To detect this periodicity, we look at a frequency spectrum generated by discrete fourier transformation (following default definition in scipy 4.5) for input sequences.

$$A_k = \sum_{m=0}^{n-1} a_m \exp\{-2\pi i \frac{mk}{n}\} \quad k = 0, \ldots, n-1 \tag{4.5}$$

First of all, we need to convert given DNA sequence into a vector of numbers. As we are interested only in particular dinucleotides, sequences are transformed into binary vectors, where ones represent positions of dinucleotides of our interest and zeros others. So for instance, sequence $AGCGGTAGG$ is translated into vector 00000100.

After that, frequency spectrum of content within 147bp long sliding window is extracted by fast fourier transformation (FFT). As we are interested only in part of this spectrum, we sum up values corresponding to period in interval from 9 to 12bp. This way, we obtain a series of numbers (number at given position corresponds to result from sliding window centered around this position), which peaks are detected by peak detection method that are marked as nucleosome centre predictions (performance shown in table 4.7).

Overall analysis of nucleosome sequences by this approach has shown a peak at 10.416bp, but it was not as strong as expected. The most probable reason for this is noise induced by frequent occurrence of AA/TT/AT/TA dinucleotides genome-wide. Some noise reduction could also be achieved by enlarging sliding window that might suit Fourier transform better.

Modification	True positives	False positives	False negatives	Sensitivity	Specificity
None	762	1,169	531	0.59	0.39
Median filter	811	1,233	482	0.63	0.40

Table 4.7: Results achieved by frequency spectrum analysis. The first line represents peak detection applied on unmodified profile, in the second case, profile was smoothed by median filter of width 11bp.

Altogether, log-ratio function and FFT scoring function perform almost identically so we come to conclusion that both key features of nucleosome sequences, k-mer differences and dinucleotide periodicity contribute to nucleosome predictions equally, at least within yeast (periodic signal is not that strong within other species).

4.3.4 Markov chain

To combine two previously described features, we turn to concept of Markov chains. More precisely, 1th order inhomogenous Markov chain, which can be defined by the dinucleotide composition at the first position $q(x_1, x_2)$, and the transitional probabilities $q(x_k|x_{k-1})$ for $k = 3, \ldots, 148$, $x_i = A/C/G/T$, $i = 1, \ldots, 148$, where k, i index the positions within a sliding window of length 148bp. These transitional probabilities were trained on nucleosome sequences of chromosomes II, III and IV and were smoothed afterwards, so that the probability for given position is equal to an average of its original value and probabilities at adjacent positions, one to the left and one to the right. This smoothing is common throughout the literature (e.g. [11], [24]) and is explained in section 3.3.1 (its effect is also shown in table 4.8).

DNA sequence is converted into numerical profile in the same fashion as in previous cases. However, Markov chain scores behave slightly different than FFT or log-ratio scores, producing weaker peaks, and thus require adjustment of signal-to-noise threshold for peak detection method or additional smoothing to achieve comparable sensitivity.

One can observe that Markov chain performs similarly to Fourier transform scoring function and log-ratios, which might indicate boundaries of predictions based on peak detection.

4.3.5 Experiment with linker length distribution

Additionally, we conducted an experiment concerning amount of information stored in linker length distribution. The question is, how many nucleosomes are we able to predict making

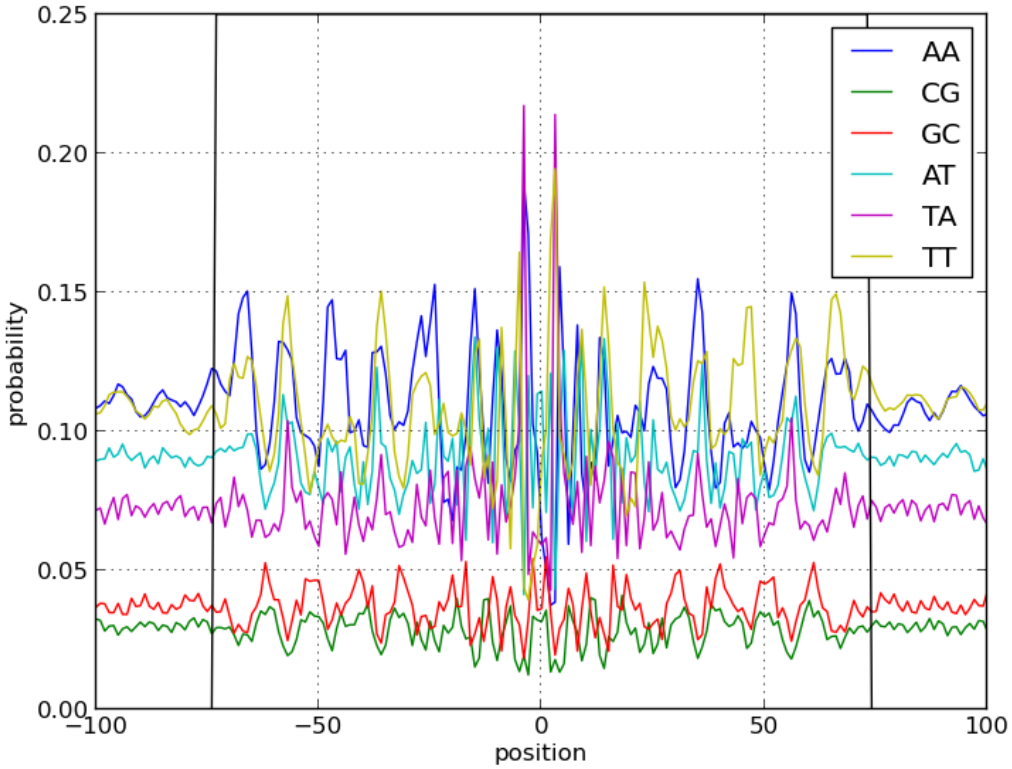


Figure 4.6: PSSM representation of Markov chain based on dinucleotide frequencies centered on ±100bp region around nucleosome centre. Boundaries of nucleosome sequence are depicted black.

Adjustment	True positives	False positives	False negatives	Sensitivity	Specificity
None*	176	247	1117	0.14	0.42
None	324	440	969	0.25	0.42
SNR.Th = 5	390	530	903	0.30	0.42
SNR.Th = 4	478	687	815	0.37	0.41
SNR.Th = 3	611	910	682	0.47	0.40
SNR.Th = 2	736	1,118	557	0.57	0.40
Smoothing	764	1,126	529	0.59	0.40

Table 4.8: Performance of Markov chain. The first line illustrates performance of Markov chain without additional smoothing of probabilities for different positions by moving average. Other lines show performance of smoothed Markov chain, uppermost with signal-to-noise threshold of peak detection set to default value of 6, followed by results attained gradually decreasing this threshold. The last line demonstrates results applying peak detection with default settings on numerical profile smoothed by median filter 11bp wide.

just random guesses based on prior knowledge about their spacing without taking into an account underlying sequence?

For this purpose, we fitted gamma distribution to lengths of linker sequences on chromosomes II, III and IV. Next, we determined positions of nucleosomes on chromosome I by generating lengths of linker sequences in-between adjacent nucleosomes according to mentioned gamma distribution. In other words, we produce pairs of numbers, representing linker and nucleosome length, the first value is a random number yielded by gamma distribution, the second one is always 147, until their sum does not exceed the length of chromosome DNA sequence.

This procedure was repeated hundred times and results are shown in table 4.9. It is important to note that one cannot compare specificity yielded by this experiment with previous ones, as they do not take into account location and distances between individual peaks, which would probably lead into smaller number of false positives.

True positives	False positives	False negatives	Sensitivity	Specificity
502	756	791	0.39	0.40

Table 4.9: Averaged results from 100 runs of linker length experiment.

One way to look at these results is to consider them as an estimation of random model performance, although it is not totally random, because we have included some information about modeled system — linker length distribution. We will come back to the topic concerning random predictions in section 5.3.3.

4.3.6 Summary

To sum up our experiments, performance of predictions based solely on peak detection, without regards to any positional information (such as distance to nearest neighbour etc.), is limited mainly by high number of false positives resulting into specificity around 0.41. However, not all of these false positives are real false positives, as we excluded from our performance measures redundant map of nucleosomes. Inclusion of this map reduce the number of false positives approximately to the half of the original value. Discussion on this topic will take place in the end of the next chapter (section 5.3.3).

With regard to nucleosome sequence properties, we conclude that k-mer distribution as well as periodical occurrences of particular dinucleotides describe nucleosome sequence preferences almost equally well in yeast and thus both of these properties should be considered by our prediction method.

Furthermore, it appears that to capture nucleosome sequence word composition preferences, statistics of dimers suffice, which is also in accord with findings in [17] where inclusion of longer sequence features resulted only in small performance improvement.

4.4 Proposed approach to nucleosome prediction

Based on previously presented observations concerning nucleosome and linker sequences, we come to proposal of our method (let us name it Nupre) for prediction of nucleosome positions in yeast inspired mainly by work on NuPop tool [24]. Similarly, we will focus directly on nucleosome positions predictions rather than just nucleosome occupancy computations. Another common aspect of these two methods is Hidden Markov model framework, which we rely on as machine learning approach for the extraction of statistical information from our dataset. We consider Hidden Markov models suitable for this kind of task especially because of its ability to capture variability in lengths of linker regions.

4.4.1 General model

General model is generalized Hidden Markov model consisting of two oscillating states — nucleosome and linker state — with explicitly modeled state duration. Each of these states is represented by slightly different submodel described in next sections. Model takes on input DNA sequence and returns the most probable path, which comprises of alternating nucleosome and linker blocks, computed by Viterbi algorithm that was modified to take into account lengths of these blocks (147bp for nucleosome state and variable length for linker state) or better state duration (see section 4.4.5).

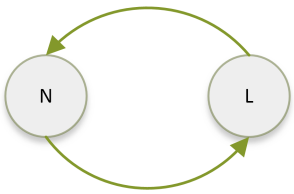


Figure 4.7: General model made of two states — nucleosome (N) and linker (L) state.

To put it simply, model proposed here is Hidden Markov model with two states and transitions between these states with probability equal to 1. Emission probabilities are generated by two submodels introduced in following sections.

4.4.2 Nucleosome model

As shown in chapter 3, majority of existing methods, which take into account 10bp periodic pattern favored by nucleosomes, model nucleosomes by PSSM or Markov chain or its other modification. These computational approaches deal with fixed probabilities for each position within matrix (or chain). By fixed, or better static, we mean that particular position has assigned specific probabilities. However, AA/TT/AT/TA dinucleotides occur every 10.5bp in general ([4, 10] or our results with FFT peak at 10.42bp) and thus vary at least between 10bp and 11bp, but these variations may be even larger (with similar assumption works e.g. [23]), and we do not consider PSSM (although smoothed by moving average) flexible enough to address this concern.

That is why we propose following two alternatives to established PSSMs — cyclic HMM and HMM profile, both designed in respect of periodic pattern and k-mer preferences of nucleosome sequences. Both alternatives were trained on DNA sequences of nucleosomes located on chromosomes II, III and IV.

Cyclic HMM

This idea comes from [3], where similar model was developed to identify periodically increased occurrence of adenine and thymine. Model consists of 10 states organized into circular topology. States are labeled from 0 to 9 allowing three transitions from each state:

- transition to the following state, so from i-th state to state $((i + 1) \mod 10)$
- transition to itself representing insertion
- and finally transition corresponding to deletion from i-th state to state $((i + 2) \mod 10)$

Model can start in each state with equal probability. Model is based on dinucleotides and their emission probabilities in states are initially set to probabilities obtained in earlier analysis (section 4.2.1), except for states 0 (or AA/AT) and 5 (GC) that are supposed to represent periodicity, where rates of AA/TT/AT/TA and GC/CG were increased respectively.

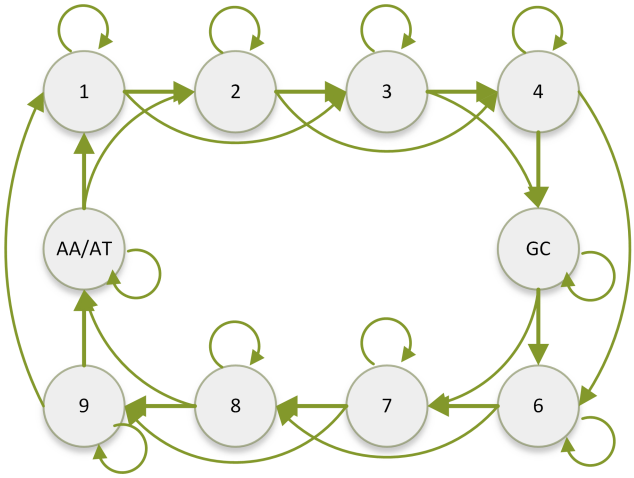


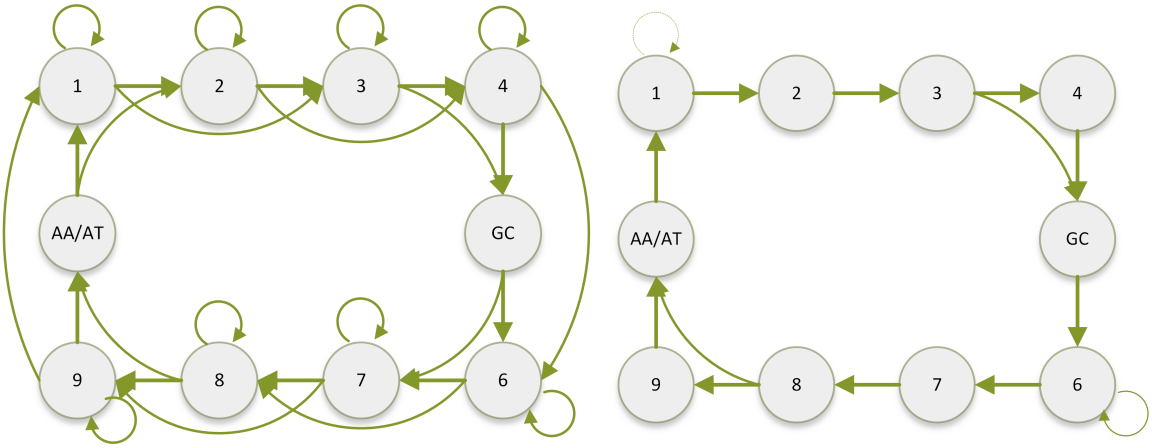
Figure 4.8: Design of cycle HMM with ten states, where states 0 and 5 were relabeled to AA/AT and CG.

For training such model, we utilize both Viterbi and Baum-Welch training procedures. We encountered the same problem with both training methods, namely huge increase in transitions to the same state in both states meant to capture periodicity — AA/AT and GC. For this reason, we also trained model having these self transitions removed (figure 4.9).

Training of latter version of model results into removal of insertion transitions, keeping only two of them, for state 1 and 6, so states directly following period states. Also only pair of deletion transitions remains, allowing to skip states preceding period states. This topology actually inspired part of next nucleosome model.

HMM profile

Closely related to PSSMs are Hidden Markov model profiles. These profile are often used for sequence alignment or representation of protein domains [18]. We propose Hidden Markov



(a) Initial model.

(b) Model trained by Viterbi algorithm.

Figure 4.9: Modified original idea with self transitions removed from periodic states. Thickness of transition lines corresponds with probability assigned to given transition.

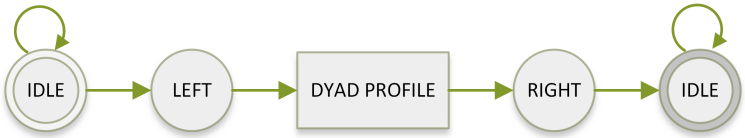


Figure 4.10: Basic elements of proposed profile topology.

model profile illustrated in figure 4.10. It is composed basically of four parts — `IDLE` states, left and right period states and profile (or PSSM) of nucleosome dyad.

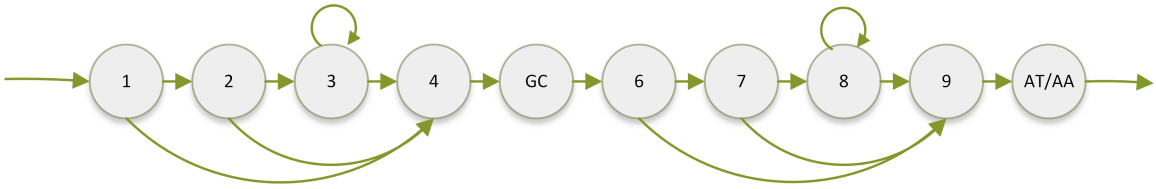
`IDLE` states are meant to capture phase of periodic signal within nucleosome sequence. Left one is followed by states representing periods of AA/AT/TA/TT upstream of nucleosome centre. Nucleosome centre, or dyad, is modeled by eight consecutive states (`DYAD PROFILE`) allowing transitions only to the following state (like Markov chain). Usage of this profile is motivated by strong presence of adenine and thymine at positions -3 and +3 relative to nucleosome centre. Central profile continues into states representing periods downstream the centre and then into last `IDLE` state.

Periodicity of AA/AT/TA/TT is described similarly as in the case of cyclic HMM trained by Viterbi algorithm (figure 4.9). In attempt to capture the periodicity of both GC/CG and AA/AT/TA/TT dinucleotides (figure 4.11a), we include two states allowing insertion (states 3 and 8) and four states allowing deletions (1, 2, 6 and 7). This stretch of states is repeated 2 to 6 times on both sides of nucleosome dyad profile, depending on desired flexibility.

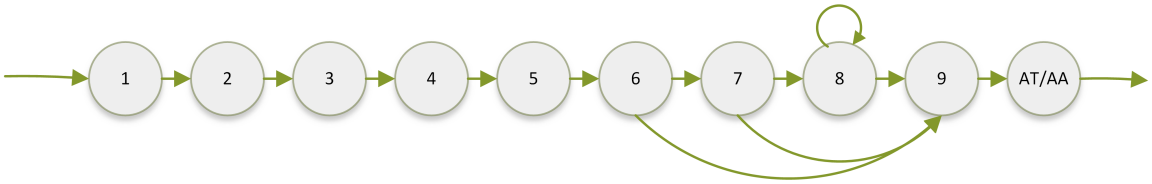
Apart from described period model, we utilize also two simpler alternatives without explicit definition of CG/GC period, where we impose more strict constraints on deviations from 10bp periodicity (figure 4.11b and 4.11c). The motivation to do so is mainly to improve results of models trained by Viterbi algorithm, which often degrades profiles to Markov chains or gets stuck increasing transition probabilities within one of insertion states.

The main idea behind this topology is to let model wind itself on 147bp long sequence, expecting it to fit better on sequences favored by nucleosomes.

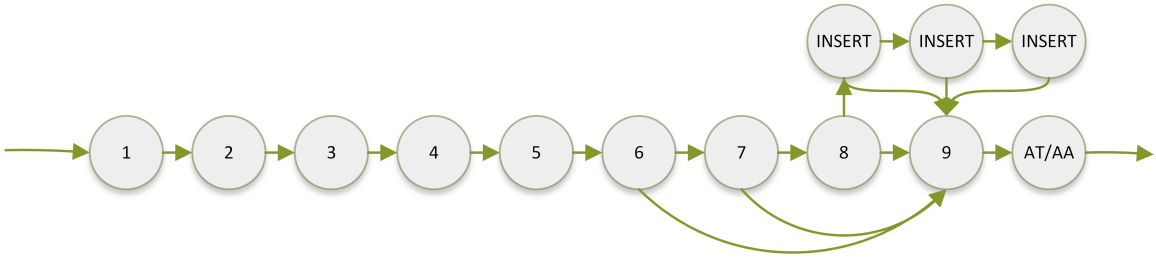
Training is a bit more complicated this time. Model starts every time in the left `IDLE`



(a)



(b)



(c)

Figure 4.11: Three alternatives for modeling one AA/AT/TA/TT period. These parts of model are inserted on the place of `LEFT` and `RIGHT` states in figure 4.10, repeated from 2 to 6 times.

state, let us label it `IDLEL` and ends in `IDLE` state to the right, `IDLER`. Initial emission probabilities were derived in the same way as in the previous case, except from emission probabilities within `DYAD PROFILE`, which were obtained from the centre alignment of nucleosome sequences. Then, we use Viterbi or Baum-Welch training procedures, to train the whole profile either on nucleosome sequences of chromosomes II, III and IV, or we partition these sequences into halves (so that we keep `DYAD PROFILE` intact) and train both halves of profile separately.

4.4.3 Linker model

Linker model is more simple then nucleosome ones. It consists of n states connected into the chain, so that there is a transition from each state to the following one. In other words, from each states i for $i < n$, there is a transition to the next state $(i+1)$ and transition to the nucleosome state of general model. Probabilities of these transitions are set accordingly to the linker length distribution. Last, n-th state has, instead of the transition to the following state, a transition to itself. This way, we can model variable length of linker regions.

Emission probabilities are shared among all the states (performs better than individual emissions for each state), so they are equal. Training of this part of the model is basic training on labeled sequences, where we take chromosome sequence of A/C/G/T and transform it into a sequence of labels $N, 1, \ldots, n$, where positions of Ns correspond to positions in the original sequence no more than 73bp away from nucleosome centres. The rest of the

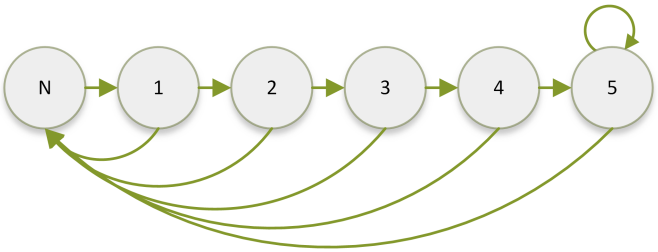


Figure 4.12: Example of linker model with 5 states ($n = 5$).

sequence is labeled according to the distance from the previous N, so that for instance sequence $N, N, -, -, -, -, N$ is labeled as $N, N, 1, 2, 3, 4, N, N$. This numbering goes up to n, if we are to insert label larger than n, then we just copy n again. Now, we count number of transitions from each state into the other (number logically correspond to the linker states and Ns to the nucleosome state) and derive transition and emission probabilities.

Trained transitional probabilities from linker to nucleosome sate can be furthermore smoothed by moving average or completely replaced by probabilities defined by gamma or normal distribution fitted onto the linker lengths, but latter generalization did not result in better performance in our experiments.

4.4.4 Assembly of submodels

Finally, an assembly of introduced parts of general model is done in straightforward way. Nucleosome state in linker model is simply replaced by one of nucleosome models (e.g. for linker model in figure 4.12, we replace state labeled as N by one of the nucleosome models) and transitions from linker states are either redirected into `IDLEL` state of HMM profile model, or branched into all states of cyclic HMM (weighted by trained probabilities for cyclic model to start in different states).

The same does not apply the other way around, because we cannot just add transition from `IDLER` state of HMM profile to the first state of linker model, as we need to ensure first that model remains in nucleosome state for 147bp.

4.4.5 State duration within HMM

Well-known weakness of Hidden Markov models is modeling of explicit state duration (e.g. [16]). In our case, this duration modeling is needed in both parts of model — in linker model to capture length distribution and in nucleosome model to ensure that predicted nucleosomes cover 147bp of DNA.

Our solution to the first problem was shown in previous section 4.4.3.

The second problem is a bit more complicated as we require from our general model to stay in nucleosome state for exactly 147bp. This problem does not occur if Markov chain (as in [24]) is used to represent nucleosome part of our model, where the length of 147bp is encoded in the number of states. Our case differs in usage of profiles for sequences shorter than 147bp, thus we propose following two extensions for traditional viterbi algorithm as our answer to state duration modeling question.

Stamp extension

The first attempt to include state duration in HMM, or at least in Viterbi algorithm, is based on time stamps. One can imagine that Viterbi algorithm works with tokens containing its probability and its path. We extend these tokens by time stamps.

`Time stamp` is tuple containing state, in which the stamp was assigned to given token, and time elapsed since this assignment. Time stamps are created by `generators`. `Generators` are transitions that create time stamps and assign them to tokens. Another extension are so called `watchouts`. They are defined by a state, a time stamp and a reaction. Basically, when a token enters a state within watchouts and this token contains time stamp defined by corresponding watchout, reaction is invoked in form of an adjustment of transition probabilities from this state. Simple example of an application of this extension is illustrated in figure 4.13.

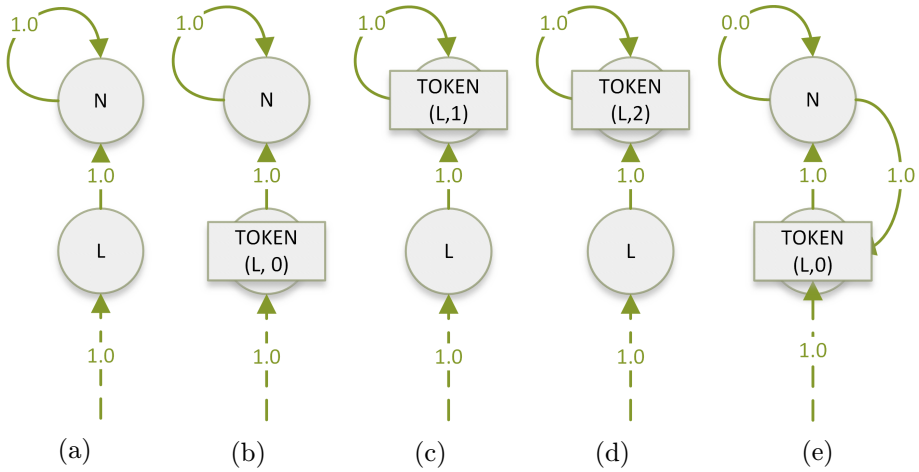


Figure 4.13: Example of stamp extension. Assume that we have two generators here: initial transition to state L and transition from state N to L (not shown because it is not part of topology definition), and one watchout for state N, which is triggered by stamp $(L, 2)$ and its reaction is a change of transition probability from N to itself to zero and from N to L to 1.0. 4.13a Topology of model with two states — L and N. 4.13b Token enters the model in state L with initial probability of 1.0. Initial transition to state L is generator, so token receives a stamp. 4.13c In the next step, token continues to state N as the only available transition from state L is to state N. Also, the counter in stamp is increased by one. 4.13d Token remains in state N following the only transition available. 4.13e Stamp $(L, 2)$ triggers reaction of watchout in state N, changing transition probabilities from state N that allows token to move to state L. As transition from N to L is generator, token is marked with new stamp.

In the case of HMM profile as submodel for nucleosome, we incorporate this extension to the model by marking all transitions from linker states to the `IDLEL` state as `generators` and adding watchout for `IDLER` state that awaits stamp $(IDLEL, 146)$ with reaction changing transition probability from `IDLER` to the first state of linker model to 1.0 (and thus transition `IDLER` to `IDLER` being set to zero). Having cyclic HMM as nucleosome submodel, this extension is integrated in similar fashion, when we tag all transitions from linker model to any state of cyclic HMM as `generators` and add watchouts for every state of nucleosome

submodel likewise.

The advantage of this extension is only relatively small increase in computational time of Viterbi decoding algorithm for the most probable path, on the other hand, we cannot guarantee finding the most probable path anymore. States with transitions to themselves (ones allowing insertions or `IDLE` states) are main bottlenecks of this approach, as they can keep fake tokens in the model. By fake, we mean tokens that are not going to make it to the last state of nucleosome submodel without exceeding 147bp limit for nucleosome blocks length. One way to deal with this issue is to add a watchout for every state with transition to itself, but then it is necessary to set stamps, which trigger these watchouts, carefully.

Nucleosome block extension

The fact that we cannot declare the path found by Viterbi decoding algorithm using the stamp extension, the most probable one, lead us to the idea of block extension.

First, we pre-compute probabilities of the most probable paths for every 147bp of an input sequence. In other words, we convert the input sequence into a numerical profile by sliding window of length 148bp (one more because we are looking at dinucleotides) moved by the step of 1bp. Afterwards, we feed values from this profile in proper time (time, when corresponding token should pass from nucleosome submodel to linker) to Viterbi decoding algorithm as probabilities generated by nucleosome submodel.

We mentioned briefly in section 4.4.2 that one of the ways to train profiles is to divide nucleosome sequence into halves and train each side of profile separately, so that we preserve `DYAD PROFILE` and guide a bit Viterbi training algorithm. Such training results in nucleosome submodel made of two parts, left and right half. In block evaluation, one can also divide sequence within sliding window into two halves, compute probabilities of each half separately with separate part of submodel and obtain final probability as product of these two halves (or sum, if we work in logarithm space). This way we can center HMM profiles to the middle of evaluated blocks, and thus avoid predictions when we match `DYAD PROFILE` correctly, but misplace nucleosome, because of `DYAD PROFILE` bias from the centre of the block.

In this case, we include in our model assembly also transitions from ending states of nucleosome submodels to the first state of linker submodel.

Apart from finding the most probable path through the input sequence, this extension allow us to add an additional smoothing of generated profile or apply a threshold that could rule out potential false predictions from computations. Unlike the previous solution, pre-computing paths of nucleosome submodel for every part of the input sequence is relatively time consuming operation.

4.4.6 Summary

Based on our observations from presented experiments concerning nucleosome sequence preferences as well as the nature of nucleosome organization within the yeast genome, we propose model that takes into account linker length distribution and dimer composition of nucleosome sequences, and extends widely used concept of PSSMs by allowing more flexibility in 10.5bp periodic pattern of dinucleotides AA/AT/TA/TT and GC/CG.

At the same time, an attempt to better describe this periodic pattern may lead into an approach too permissive towards the structure of DNA sequence resulting in higher number of false positive predictions. On the other hand, replacing Markov chains with HMM imparts additional information about alignment, or the optimal path of nucleosome

submodel through the underlying sequence. Especially positions of `DYAD PROFILE` might be of great interest.

Chapter 5

Testing and results

This chapter describes test methodology and results achieved by our approach to nucleosome prediction. We test several versions of model varying in the length or additional smoothing of its linker part and in lengths of HMM profiles, techniques of their training or evaluation.

The performance is compared with *reference model*, which consists of Markov chain for nucleosome representation (the same we used in section 4.3.4).The *reference model* should give us a brief overview of potential improvement induced by usage of HMM profiles instead of well-established PSSMs.

5.1 Test design

In addition to testing our model on the entire chromosome I, we prepared three other test sets. Each set was designed to address different part of our model.

The first one is focused on nucleosome part of the model omitting any linker length distribution properties. Set consists of 200bp long DNA stretches that are occupied by single nucleosome. We picked 200 such regions, 150 of them randomly, 25 with nucleosomes exhibiting high log-ratios scores (see section 4.3.2) and other 25 low. Model was simplified by removal of its linker part, which was replaced by two states (figure 5.1) — one in front of the nucleosome model, another behind — having the same emission probabilities as states within original linker model.

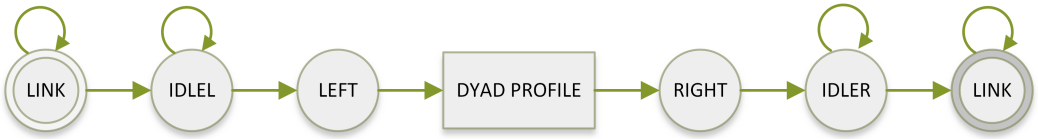


Figure 5.1: Modification of original model for single nucleosome predictions.

The second set contains eight regions of 5,000bp with high variance in lengths of linker DNA in-between nucleosomes. Elements of this set cover sparsely spaced short gene-coding regions, retranspozons or dubious gene predictions. The purpose of this set is to explore behaviour of the linker submodel.

The third one on the contrary include eight regions with low variance in linker DNA lengths. These regions are mostly gene-coding regions encoding one or several densely positioned genes, which explains low variance in the phasing of nucleosomes (section 2.1.2).

Low variance in linker lengths should test mainly nucleosome part of the model and model's ability to predict unknown number of nucleosomes (as opposed to the first test set).

Label	Position	Notes
Low variance		
reg_low_0	chrI:14,650-19,650	None
reg_low_1	chrI:53,450-58,450	dense coding region
reg_low_2	chrI:61,400-66,400	coding region
reg_low_3	chrI:94,550-99,550	coding region
reg_low_4	chrI:133,850-138,850	several coding regions
reg_low_5	chrI:143,200-148,200	coding region
reg_low_6	chrI:153,500-158,500	several coding regions
reg_low_7	chrI:177,300-182,300	coding region
High variance		
reg_high_0	chrI:29,600-34,600	two protein coding regions
reg_high_1	chrI:66,861-71,861	three sparse coding regions
reg_high_2	chrI:78,350-83,350	dense protein coding region
reg_high_3	chrI:136,980-141,980	sparse coding region
reg_high_4	chrI:160,000-165,000	two restranspozons
reg_high_5	chrI:187,700-192,700	sparse coding region
reg_high_6	chrI:207,000-212,000	two dubious ORFs
reg_high_7	chrI:215,550-220,550	hypotetical coding regions

Table 5.1: Positions and descriptions of chosen regions with the lowest and the highest variance in linker lengths on chromosome I. Regions are also enclosed as supplementary material in fasta format.

5.2 Testing and fine-tuning

5.2.1 Single nucleosome prediction

The goal of this test is to determine performance and behaviour of nucleosome part of the model, knowing that there is only one nucleosome to predict. We tested profiles of different lengths varying from 2 periods on each side of the `DYAD PROFILE` up to 6 using both Viterbi and Baum-Welch training algorithm. Also both of our duration extensions were put to the test.

In this case, we used different benchmarking technique, when instead of looking at numbers of true positives or negatives we consider the distance of true nucleosome centre from the predicted one as metric.

Our *reference model* based on Markov chain outperformed all HMM profiles with an average distance of predicted centres from true ones 13.34bp and median 7.5bp. Interestingly, the weakest performance could be observed at the part of our test set with nucleosomes having high log-ratio scores (see table 5.2).

The best results among cyclic HMM submodels were achieved by model trained by Baum-Welch algorithm relying on our block extension of HMM. In general, submodels trained by Baum-Welch method performed slightly better than theirs alternatives trained by Viterbi algorithm. The best among HMM profiles was one with 4 repeats of period,

Nucleosome model		Markov chain	Cyclic HMM	HMM profile	halves HMM profile
High log-ratios	median	14.0	14.0	17.0	14.0
	average	15.08	15.12	18.24	14.56
Low log-ratios	median	9.0	17.0	12.0	11.0
	average	10.88	18.44	17.2	11.92
Overall	median	8.0	15.0	15.5	12.0
	average	13.01	19.51	19.34	15.9

Table 5.2: The best performance achieved by representatives of different nucleosome submodels. Values represent either average or median distance of predicted nucleosomes centres from the real centres in base pairs. High log-ratios stand for 25 regions in our test set with strong peaks in log-ratio scores, low log-ratios on the other hand with very low, actually lowest, log-ratio scores (section 4.3.2). Overall rows correspond to the performance on the entire test set consisting of 200regions occupied by single nucleosome. Column with label halves HMM profile shows performance of nucleosome mode working with block extension, when halves of block are considered separately (see section 4.4.5).

without GC/CG explicitly specified and with restricted number of insertions (figure 4.11c), at each side of `DYAD PROFILE` trained by Baum-Welch technique. Better than HMM profile as a whole performed profiles divided into halves, and thus keeping `DYAD PROFILE` in the middle of blocks during Viterbi decoding algorithm, capturing 3 period repeats around `DYAD PROFILE` defined the same way as in the case of best performing HMM profile.

While block extension of Viterbi decoding algorithm yielded average distances from the real nucleosome centres around 20-22bp for examined profiles, stamp extension failed to deliver similar results with averages increased to 30bp.

Despite the superior performance of *reference mode*, the flexibility encoded in HMM profile provides an additional information about an alignment of its `DYAD PROFILE`. To take an advantage of this supplementary knowledge, we come back to sliding windows. Moving such window of length 147bp with 1bp step along an input sequence and obtaining optimal path for every part of the sequence covered results in clusters of `DYAD PROFILE` alignments — positions, where Viterbi algorithm placed `DYAD PROFILE` (illustrated in figure 5.2).

So, in addition to the most probable path of model for given input, we also get positions of putative nucleosomes that correspond to nucleosomes centres, where model for single nucleosome predictions failed, to be more precise, we observed centre state (centre state of `DYAD PROFILE`) cluster located within 10bp from nucleosome centre in almost 60% of examined regions.

Another aspect to take into consideration is redundant set of nucleosomes. When we include these nucleosomes in our performance assessment, average distance to the closest nucleosome centre, either from unique or redundant set, in the case of the best performing HMM profile decreases from 19.36bp down to 8.99bp with median of 5bp.

To sum up single nucleosome predictions, all examined models performed well considering commonly used tolerance for true positive match being 35bp. From this point of view, predictions were really accurate with specificity around 84%, but prediction of known number, especially when there is only one, of nucleosomes is large simplification of real problem. Nevertheless, this testing narrowed selection of nucleosome submodels for further evaluation and offered initial comparison between proposed approach and conventional PSSMs.

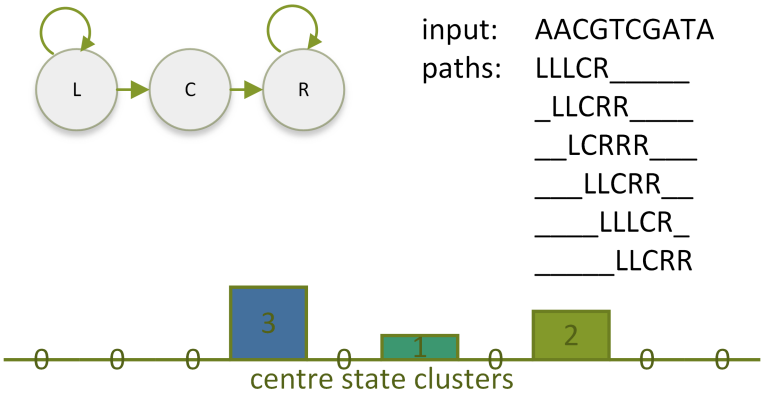


Figure 5.2: Simplified example of obtaining supplementary information from HMM profile alignments. Consider model shown in the top left corner consisting of three states — L, C and R — starting in state L and ending in state R. We are interested in positions of C (representing `DYAD PROFILE` from our profiles) within most probable paths through sliding window of length 5bp. These paths for given input sequence are shown in the right corner. Profile created regarding to these positions is depicted at the bottom, when number for particular position correspond to the number of times state C was aligned to the position.

5.2.2 Prediction in low and high linker variance regions

Turning to linker part of the model, we make prediction task more difficult by applying model on longer regions without any knowledge of number of nucleosomes to match. The main purpose of this test is to find an optimal number of states included in linker part of the model and method to determine transition probabilities from linker to nucleosome. In this and forthcoming tests, we restrict our comparisons to the best performing nucleosome submodels from the previous section.

Reference model

First of all, we examine effects of an additional smoothing imposed on transitional probabilities from linker states to nucleosome. To obtain some basic comparison we apply several smoothing methods on model based on Markov chain as nucleosome submodel and 60 states long linker part. Results are shown in table 5.3 and do not indicate any improvement in performance.

Next, we move to the number of linker states. Our goal is to determine effective number of linker states, a number of states above which performance does not get better or degrade. To obtain this number, we vary the amount of states included in the linker submodel from 40 up to 200. Having, for example, 40 states within linker submodel does not mean that we do not allow longer linkers, but that linkers longer than 40bp are treated equally in terms of transition probabilities to the nucleosome state or in other words we omit linker length distribution for linkers with lengths above the number of states.

Table 5.4 shows that performance peaks at previously studied linker submodel with 60 states. Apart from that, one can observe worse results in regions with high variance in linker lengths, when the model places false nucleosomes along the long spaces in nucleosome phasing.

Smoothing	None		Moving avg.		Gamma distr.		Normal distr.	
	low	high	low	high	low	high	low	high
True positives	135	91	125	93	98	61	103	75
False positives	85	112	97	114	81	81	79	81
False negatives	113	99	123	97	150	129	145	115
Sensitivity	0.54	0.48	0.50	0.49	0.40	0.32	0.42	0.39
Specificity	0.61	0.45	0.56	0.45	0.55	0.43	0.57	0.48

Table 5.3: Methods for determination of transition probabilities and their effect on performance using linker submodel with 60 states and Markov chain. Low and high columns for each method show results from low and high variance linker regions respectively. Moving avg. stands for smoothing transition probabilities with moving average of 3, in the case of gamma distr. we replaced trained probabilities with probabilities given by gamma distribution fitted to linkers in the training set, the same applies for normal distribution in the last column.

Number	40		60		80		100		150		200	
	low	high	low	high	low	high	low	high	low	high	low	high
True positives	122	92	135	91	126	90	127	83	128	81	130	79
False positives	100	113	85	112	98	109	99	112	98	102	95	100
False negatives	126	98	113	99	122	100	121	107	120	109	118	111
Sensitivity	0.49	0.48	0.54	0.48	0.51	0.47	0.51	0.44	0.52	0.43	0.52	0.42
Specificity	0.55	0.45	0.61	0.45	0.56	0.45	0.56	0.43	0.57	0.44	0.58	0.44

Table 5.4: Impact of the number of states on the performance.

Proposed submodels

With respect to the optimal configuration of linker model from our *reference model*, we move to evaluation of proposed nucleosome submodels in cooperation with linker part. Table 5.5 presents initial results and we observe performance drop on both, sensitivity and specificity.

Submodel	Cyclic HMM		HMM profile		halves HMM profile	
	low	high	low	high	low	high
True positives	124	91	94	82	94	68
False positives	139	172	144	155	97	109
False negatives	124	99	154	108	154	122
Sensitivity	0.5	0.48	0.38	0.43	0.38	0.36
Specificity	0.47	0.35	0.39	0.35	0.49	0.38

Table 5.5: Results from tests of different nucleosome submodels with 60 states long linker using block extension of the Viterbi algorithm.

Profile submodels look to to be more permissive than desired and thus an additional restrictions are needed in finding the most probable path. Apart from that, linker model

apparently fails to insert longer regions into predicted paths, as specificity is always lower in regions with high linker length variance.

One way to get rid of redundant predictions is to impose a lower limit on the acceptable nucleosome predictions. To derive suitable threshold, nucleosomes on chromosome III, as part of our training set, were rated by probabilities (to be more precise, log-likelihoods as we work in log-space) of the most probable paths trough given nucleosome submodel. The assessment of these ranks is shown in figure 5.3. Different classes of ranks are completely overlapping and, despite the fact that distribution considering only unique nucleosomes is shifted bit towards higher scores compared to the other two classes, an addition of thresholds leads to equal drops in both true and false positive predictions.

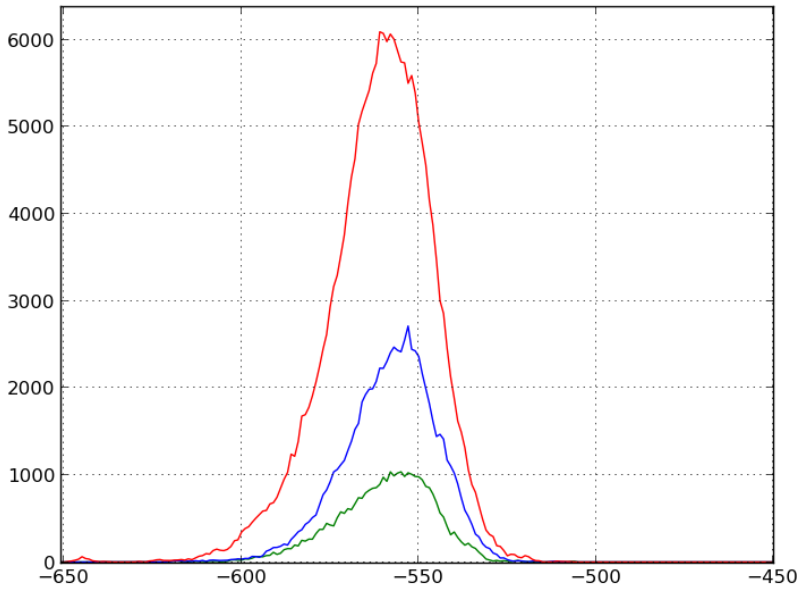


Figure 5.3: Observed log-likelihoods were distributed into three classes, one for scores yielded by blocks centered around unique nucleosome centres with ±10bp tolerance, depicted in green, blue line corresponds to redundant nucleosomes and red one stands for scores observed outside previous two classes. On x-axis we have log-likelihoods of the most probable paths found by model throughout the chromosome III, y-axis represents how many times was the log-likelihood observed for given class.

Another option examined was an usage of scaling factor, when the probability emitted by nucleosome submodel was scaled down by a given number, but again no considerable improvement was observed.

The last attempt to improve performance of HMM profile model on this testing set involves additional smoothing of pre-computed nucleosome block likelihoods. This smoothing is carried out by median filter with kernel size 11, which proved to return better results in peak detection methods (section 4.3.1). The improvement was achieved in both sensitivity (from 0.38 in low and 0.43 in high variance regions, to 0.44 and 0.49 respectively) and specificity (from 0.39 and 0.35 to 0.45 and 0.38).

5.3 Results

5.3.1 Large-scale predictions

To complete our evaluations, we conduct the last experiment including large-scale predictions. This time, the model is to label entire sequence of chromosome I at once. The calculations of Viterbi path are more time consuming in this case, so we also examine our time stamp heuristic that is multiple times faster than block evaluation.

Method	Markov chain	Cyclic HMM	HMM profile	HMM profile stamp	halves HMM profile
True positives	643(1007)	564(1089)	607(1070)	485(909)	485(833)
False positives	623(259)	951(426)	835(372)	769(345)	623(275)
False negatives	650(3661)	729(3871)	686(3841)	808(4424)	808(4,493)
Sensitivity	0.50(0.22)	0.44(0.22)	0.47(0.22)	0.38(0.17)	0.38(0.16)
Specificity	0.51(0.80)	0.37(0.72)	0.42(0.74)	0.39(0.72)	0.44(0.75)

Table 5.6: Performance achieved analysing chromosome I. In brackets are results considering redundant dataset.

Once again *reference mode* overperformed other configurations. Noteworthy is the performance of model with HMM profile as nucleosome state, which yielded comparable results with or without additional smoothing. Nevertheless, the low specificity is an issue. These low values and thus high numbers of false positives can be partly explained by redundant set of nucleosomes, which we did not take into account up to this point. Table 5.6 illustrates how many false positives are not truly false positives, as they corresponds to some other position of the original nucleosome.

5.3.2 Comparison with NuPoP

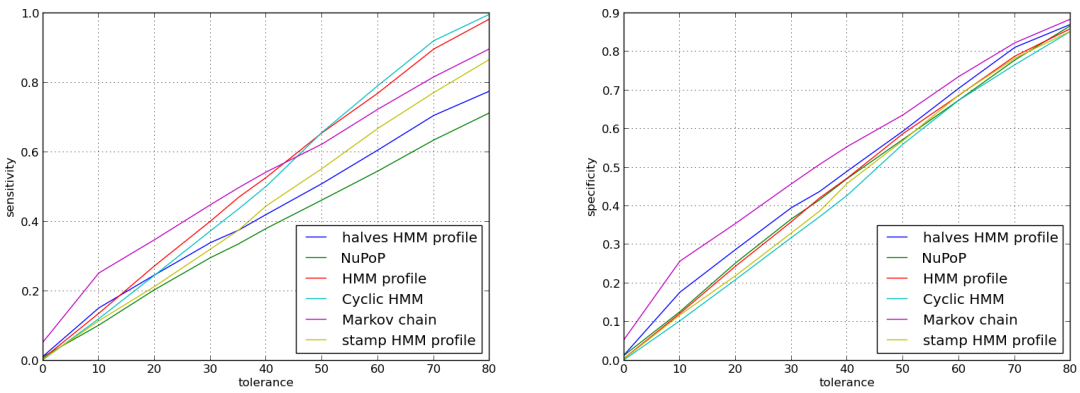
Moreover, we compare presented models with NuPoP [24], which is based on similar principles of duration HMM. As we did not manage to obtain the dataset used for training and testing this tool, our comparison is going to be a bit biased. That is why results of our *reference model* are shown as well, which conceptually differs with NuPoP only slightly in the linker part of the model, and thus provides the closest approximation of NuPoP trained on our dataset available.

Furthermore, two models are available within NuPoP for nucleosome prediction — one based on the 1st order Markov chain and the second one on the 4th order Markov chain as nucleosome representation. To avoid overfitting as much as possible, we choose the 1st order Markov chain in the most of our comparisons, but in general, both models performed equally well.

Tables 5.7 and 5.8 show a behaviour of different models in regard to individual sequences of our test sets based on variance in linker lengths within given region. Results achieved vary a lot between used models as well as given sequences, allowing only the confirmation of the significantly better performance of trained Markov chain compared to the other configurations. Interestingly, it outperforms NuPoP in the majority of examined regions especially in terms of sensitivity, which implies relatively large differences between used training sets.

Method		NuPoP	Markov chain	Cyclic HMM	profile HMM	halves HMM profile
reg_high_0	Sn	0.27	0.54	0.72	0.5	0.45
	Sp	0.26	0.48	0.48	0.37	0.43
reg_high_1	Sn	0.19	0.48	0.24	0.62	0.29
	Sp	0.18	0.45	0.15	0.43	0.33
reg_high_2	Sn	0.42	0.38	0.42	0.38	0.54
	Sp	0.48	0.36	0.34	0.31	0.58
reg_high_3	Sn	0.36	0.36	0.28	0.52	0.24
	Sp	0.41	0.32	0.21	0.42	0.25
reg_high_4	Sn	0.35	0.74	0.43	0.26	0.48
	Sp	0.36	0.59	0.30	0.2	0.44
reg_high_5	Sn	0.35	0.43	0.57	0.74	0.22
	Sp	0.30	0. 67	0.39	0.55	0.31
reg_high_6	Sn	0.25	0.5	0.62	0.42	0.29
	Sp	0.33	0.43	0.45	0.32	0.28
reg_high_7	Sn	0.19	0.42	0.54	0.50	0.35
	Sp	0.22	0.39	0.42	0.43	0.41
Overall	Sn	0.30	0.48	0.48	0.49	0.36
	Sp	0.32	0.45	0.35	0.38	0.38

Table 5.7: Detailed comparison of NuPoP and developed models on regions with high variance in linker lengths.



(a) sensitivity

(b) specificity

Figure 5.4: The summary of the performance of proposed nucleosome submodels and NuPoP in terms of specificity and sensitivity with decaying tolerance — allowed distance from the real nucleosome centre within which are predicted nucleosomes still considered true positives.

5.3.3 Discussion

A large number of false positives, which can be partly explained by redundant map of nucleosomes, remains an issue also after the incorporation of information about positioning

Method		NuPoP	Markov chain	Cyclic HMM	profile HMM	halves HMM profile
reg_low_0	Sn	0.29	0.74	0.42	0.42	0.48
	Sp	0.60	0.79	0.41	0.42	0.63
reg_low_1	Sn	0.50	0.43	0.5	0.5	0.5
	Sp	0.64	0.44	0.42	0.47	0.56
reg_low_2	Sn	0.44	0.59	0.66	0.38	0.44
	Sp	0.66	0.68	0.64	0.39	0.54
reg_low_3	Sn	0.37	0.67	0.47	0.30	0.23
	Sp	0.55	0.69	0.42	0.29	0.28
reg_low_4	Sn	0.31	0.44	0.59	0.34	0.34
	Sp	0.63	0.5	0.58	0.35	0.44
reg_low_5	Sn	0.41	0.50	0.41	0.50	0.50
	Sp	0.72	0.57	0.39	0.50	0.64
reg_low_6	Sn	0.25	0.44	0.56	0.59	0.31
	Sp	0.36	0.52	0.55	0.61	0.48
reg_low_7	Sn	0.35	0.55	0.39	0.52	0.23
	Sp	0.48	0.63	0.36	0.53	0.35
Overall	Sn	0.36	0.54	0.50	0.44	0.38
	Sp	0.57	0.61	0.47	0.45	0.49

Table 5.8: Detailed comparison of NuPoP and developed models on regions with low variance in linker lengths.

of nearby nucleosomes by dynamic programming technique.

The fuzziness of nucleosome positions, explaining more than a half of false positive predictions, is not encoded within our model, which is able to capture nucleosomes only in one static moment (snapshot) and not their fluctuations. However, going back to log-ratios (section 4.3.2), peaks of this scoring function were usually broaden around fuzzy nucleosomes from the redundant map, so it should be possible to predict prediction and trajectory of these 'floating' nucleosomes should be possible based on the DNA sequence.

Another problem we observed is the prediction of longer linker regions. Proposed models place several false nucleosomes within these regions implying that models are not able to discriminate between linker and nucleosome sequences sufficiently.

Random performance

As nucleosome prediction methods often exhibit low specificity and sensitivity around 0.5, the question concerning their addition to the information stored in the linker length distribution and the oscillating nature of nucleosomes and linkers arise. We showed earlier that we are able to guess nucleosome positions just by random picks of linker lengths with specificity and sensitivity around 0.4.

To demonstrate the contribution of the trained submodels to the information the lengths of nucleosome repeats, we randomized an input DNA sequence and predicted nucleosomes by our *reference model* in regard to their original positions within chromosome I. Predictions yielded 515 true positives that is 80% of true positive matches on the original sequence.

Chapter 6

Conclusion

In general, genome-wide nucleosome predictions suffer from the high number of false positive predictions as well as relatively low sensitivity ranging around 50%. The real question is, how many nucleosomes is it even possible to accurately predict, as they usually cover around 80% of genome, and thus any dataset will be rich at least in the variety within nucleosome sequences.

Our proposal of HMM profiles, to better describe periodical pattern of particular dinucleotides in nucleosome sequences, incorporated within Hidden Markov model considering state duration did not yield desired improvement in comparison to existing methods. The flexible nature of proposed profiles provides an additional information about alignment of such profiles (centre clustering), but also introduces an undesired noise into predictions.

The time stamp extension heuristic (section 4.4.5) was introduced into the concept of Viterbi algorithm to reduce increased time consumption related to the inclusion of state duration modeling within HMMs.

Moreover, published dataset [4] was analyzed in respect to different features of nucleosome sequences. These experiments might be considered as sort of prediction techniques, but the issue with low specificity (approximately 40%), and thus high number of false positives, remains.

Furthermore, we presented interesting observations about the symmetry encoded within nucleosome sequences (section 4.2.1) and indicated that polyA:T tracts are actually quiet often occupied by nucleosomes (section 4.2.1) in our dataset as opposed to the general assumption that sequences containing polyA:T are disfavored by nucleosomes.

6.1 Further work

In the future, performance might be improved by an addition of more pre- or post-processing steps, such as omitting regions with long tracts of adenine and thymine (although this does not have to be very beneficial as demonstrated in section 4.2.1), different smoothing techniques applied on numerical landscape of nucleosome submodel etc. Also retraining model in respect to k-mers with $k > 2$ could lead to better results.

Proposed approach does not take into consideration statistical positioning phenomenon [14].

One possibility is to dynamically adjust threshold applied in our block extension, when we would start with high threshold, allowing us to filter out false positive predictions in long linker regions, then after finding suitable +1 nucleosome, we could lower the threshold

and gradually increase it, together with the increasing distance from predicted +1, back to its original values.

Another way to do so is to include some confidence measure into our general model and extend it with one more state, let us label it U as undefined. When confidence level drop below defined threshold, model will move to state U in which it looks for the next anchor point. This search might be performed by other technique than HMM profiles, which might be trained directly on TSS regions to capture -1 and +1 nucleosomes (or more, as +1 nucleosome might exhibit actually lower sequence positioning signal because of TSS often lie on the boundaries of this nucleosome and thus weaker bonding may be needed) and nucleosome depleted region in-between.

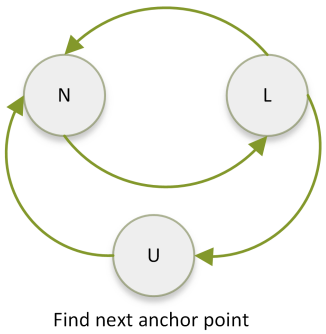


Figure 6.1: Schematics of possible extension of general model.

Furthermore, the model could be modified in such manner that it will be able to predict the fuzziness in nucleosome positioning. So instead of fixed length of predicted nucleosomes, we would consider also longer blocks, which might be achieved by gathering additional information from redundant map (but on the other hand, it means sacrificing information about linker length distribution).

Bibliography

[1] Anthony T. Annunziato. DNA packaging: Nucleosomes and Chromatin. online http://www.nature.com/scitable/topicpage/dna-packaging-nucleosomes-and-chromatin-310, 2013.

[2] Gaurav Arya, Arijit Maitra, and Sergei A. Grigoryev. A structural perspective on the where, how, why, and what of nucleosome positioning. *Journal of Biomolecular Structure & Dynamics*, 27:803–820, 2010.

[3] Pierre Baldi, Søren Brunak, Yves Chauvin, and Anders Krogh. Naturally occurring nucleosome positioning signals in human exons and introns. *Journal of Molecular Biology*, 263, 1996.

[4] Kristin Brogaard, Liqun Xi, Ji-Ping Wand, and Jonathan Widom. A map of nucleosome positions in yeast at base-pair resolution. *Nature*, 486:496–501, 2012.

[5] G. Chevereau, L. Palmeira, C. Thermes, A. Arneodo, and C. Vaillant. Thermodynamics of intragenic nucleosome ordering. *Physical Review Letters*, 103, 2009.

[6] Pan Du, Warren A. Kibbe, and Simon M. Lin. Improved peak detection in mass spectrum by incorporating continouos wavelet transform-based pattern matching. *Bioinformatics*, 22:2059–2065, 2006.

[7] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[8] I. Gabdank, D. Barash, and Edward N. Trifonov. Nucleosome DNA bendability matrix (C. elegans). *Journal of Biomolecular Structure & Dynamics*, 26:403–411, 2009.

[9] Vishwanath R. Iyer. Nucleosome positioning: bringing order to the eukaryotic genome. *Trends in Cell Biology*, 22:250–256, 2012.

[10] Noam Kaplan, Irene Moore, Yvonne Fondufe-Mittendorf, Andrea J. Gossett, Desiree Tillo, Yair Field, Timothy R. Hughes, Jason D. Lieb, Jonathan Widom, and Eran Segal. Nucleosome sequence preferences influence in vivo. *Nat Struct Mol Biol*, 17:918–922, 2010.

[11] Noam Kaplan, Irene K. Moore, Yvonne Fondufe-Mittendorf, Andrea J. Gossett, Desiree Tillo, Yair Field, Emily M. LeProust, Timothy R. Hughes, Jason D. Lieb, Jonathan Widom, and Eran Segal. The DNA-encoded nucleosome organization of a eukaryotic genome. *Nature*, 458:362–366, 2009.

[12] Roger D. Kornberg and Yahll Lorch. Twenty-five years of the nucleosome, fundamental particle of the eukaryote chromosome. *Cell*, 98:285–294, 1999.

[13] Alexander V. Lukashin and Mark Borodovsky. GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Research*, 26:1107–1115, 1998.

[14] Travis N. Mavrich, Ilya P. Ioshikhes, and Bryan J. Venters.

[15] Heather E. Peckham, Robert E. Thurman, Yutao Fu, John A. Stamatoyannopoulos, William Stafford Noble, Kevin Struhl, and Zhiping Weng. Nucleosome positioning signals in genomic DNA. *Genome Research*, 17:1170–1177, 2007.

[16] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286.

[17] Sheila M. Reynolds, Jeff A. Bilmes, and William Stafford Noble. Learning a weighted sequence model of the nucleosome core and linker yields more accurate predictions in Saccharomyces cerevisiae and Homo sapiens. *PLoS Computational Biology*, 6, 2010.

[18] Erik L. L. Sonnhammer, Sean R. Eddy, Ewan Birney, Alex Bateman, and Richard Durbin. Pfam: Multiple sequence alignments and hmm-profiles of protein domains. *Nucleic Acids Research*, 26:320–322, 1998.

[19] Desiree Tillo and Timothy R. Hughes. G+C content dominates intrinsic nucleosome occupancy. *BMC Bioinformatics*, 10, 2009.

[20] Michael Y. Tolstorukov, Vidhu Choudhary, Wilma K. Olson, Victor B. Zhurkin, and Peter J. Park. nuScore: a web-interface for nucleosome positioning predictions. *Bioinformatics*, 24:1456–1458, 2008.

[21] Martin Tompa, Nan Li, Timothy L. Bailey, George M. Church, Bart De Moor, Eleazar Eskin, Alexander V. Favorov, Martin C. Frith, Yutao Fu, W. James Kent, Vsevolod J. Makeev, Andrei A. Mironov, William Stafford Noble, Giulio Pavesi, Graziano Pesole, Mireille Régnier, Nucolas Simonis, Saurabh Sinha, Gert Thijs, Jacques van Helden, Mathias Vandenbogaert, Zhiping Weng, Christopher Workman, Chun Ye, and Zhou Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23:137–144, 2005.

[22] The Open University. Nucleic acids and chromatin. online http://www.open.edu/openlearn/science-maths-technology/science/biology/nucleic-acids-and-chromatin/content-section-7.3.2, June 2013.

[23] Qinqin Wu, Jiajun Wang, and Hong Yan. Prediction of nucleosome positions in the yeast genome based on matched mirror position filtering. *Bioinformation*, 3:454–459, 2009.

[24] Liqun Xi, Yvonne Fondufe-Mittendorf, Lei Xia, Jared Flatow, Jonathan Widom, and Ji-Ping Wang. Predicting nucleosome positioining using a duration Hidden Markov Model. *BMC Bioinformatics*, 11, 2010.

[25] Yongqiang Xing, Xiujuan Zhao, and Lu Cai. Prediction of nucleosome occupancy in saccharomyces cerevisiae using position-correlation scoring function. *Genomics*, 98:359–366, 2011.

[26] Chao Yang, Zengyou He, and Weichuan Yu. Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. *BMC Bioinformatics*, 10, 2009.

[27] Guo-Cheng Yuan and Jun S. Liu. Genomic sequence is highly predictive of local nucleosome depletion. *PLoS Computational Biology*, 4:164–174.

Appendix A

Structure of enclosed digital material

Enclosed with this thesis is also a digital material containing a command line application NuPre for prediction of nucleosome positions as well as other scripts used during presented experiments. To use this application as well as other scripts, python interpreter is required together with other libraries mentioned in the beginning of the chapter 4. Here we give an overview of the folder structure of enclosed material, to make it easier to orient. More examples should be available at http://bioware.fit.vutbr.cz/mediawiki/index.php/Nucleosome_prediction.

- `/`
  - `data`
    - `browser`
      - `*.bed` ..... bed files suitable for viewing in genome browsers containing information about nucleosome directionality
    - `datasets`
      - `*.dts` ..... used datasets
    - `regions`
      - `*.fasta` ..... test regions with low and high linker variance
    - `tables`
      - `*.tbl` ..... various tables with kmer statistics
    - `S288C_reference_genome_R61-1-1_20080605.fasta` ..... reference genome
  - `doc`
    - `index.html` ..... entry file for the documentation generated by `epydoc` package
  - `src`
    - `analysis`
      - `dataset.py` ..... module with structures for manipulation with dataset
      - `experiments.py` .. some experiments conducted in this thesis accessible from `other.py`
      - `general.py` ..... definitions of global variables
      - `prepare_sets.py` ..... helper functions to work with nucleosome map
      - `scoringfunction.py` ..... definition of scoring functions
      - `slidingwindow.py` ..... sliding window class definitions
    - `models`
      - `linker` ..... folder with saved linker models

- `general_*.hmm` ..... general models for nucleosome prediction with specified nucleosome submodel
- `single_*.mm` ..... models for single nucleosome predictions
- `*.hmm` ..... nucleosome submodels

`prediction`

- `hmm_2.py` ..... implementation of HMM framework
- `linker.py` .....
- `predictor.py` ..... module containing some helper variables and functions
- `tester.py` ..... script used for testing
- `trainer.py` ..... defines training routines

`example.fasta` ..... example of input data

`nupre.py` ..... entry script for NuPre application

`other.py` ..... script to access some other experiments

`peak.R` ..... script for peak detection

`README` ..... examples of nupre usage